UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Reliable Communications within Cyber-Physical Systems Using the Internet (RC4CPS)

Mohammad Elattar

Dissertation
in Computer Science

Faculty of Electrical Engineering,
Computer Science and Mathematics

University of Paderborn

in partial fulfillment of the requirements for the degree of

Doktor der Naturwissenschaften

(Dr. rer. nat.)

Lemgo, June 2018

Abstract

An important requirement to realize cyber-physical systems (CPSs) in critical infrastructures such as power grids is communication reliability where such reliability is usually measured in terms of communication service unavailability. With this regard, applications proposed for smart grids have reliability requirements of 99-99.9999%. To achieve this, most power utilities rely on dedicated networks and/or leased lines. The alternative for such solutions is the Internet which represents a global, cost-effective network for CPSs that span large geographical areas. Unfortunately, the reliability of today's Internet is inadequate and varies over time. A widely adopted approach in other domains to enhance reliability utilizes mainly redundancy in terms of communication paths and transmitted data. However, this requires knowledge about the topology to ensure disjointness of used paths, which is difficult in case of the Internet. Even with such knowledge, most of the available multipath (MP) communication protocols are throughput-oriented or proposed for dedicated networks and, therefore, cannot be utilized directly. Nevertheless, MP communication is still expected to improve the communication reliability of the Internet.

In this dissertation, data duplication and dynamic MP selection during runtime when using multiple end-to-end (e2e) paths are proposed to improve the communication reliability for Internet-based CPSs. With this regard, the problem of paths selection is formulated as an optimization problem to select the minimum number of e2e paths and limit the redundant data by the needed reliability. The multiple e2e paths are realized using different access internet service providers and MP communication protocols. In addition, real world measurements to investigate the reliability gains of MP communication in the Internet were conducted. The obtained results proved the existence of e2e paths that traverse completely different networks and, consequently, are likely to be disjoint. They also showed that the concurrent unavailability of different subsets of paths with two and three paths was 0%. Those results motivated proposing the Reliable Multipath Communication for Internet-based CPSs (RC4CPS) approach. It is an e2e approach that utilizes the inherent redundancy of the Internet and the concept of MP communication protocols to improve reliability. It also provides online monitoring and dynamic MP selection that considers the diversity and unavailability probability of e2e paths to maximize the reliability gains. RC4CPS was first implemented in MATLAB for initial evaluations and, then, using the iPRP (Parallel Redundancy Protocol for IP Networks) MP transport protocol. The resulting protocol, called iPRP-RC4CPS, incorporates the RC4CPS features and extends the original iPRP implementation, proposed for dedicated WAN networks, to support the Internet. The evaluation results carried out using both implementations of RC4CPS in the Internet indicated the ability of iPRP-RC4CPS to achieve 0% unavailability while selecting the minimum number of e2e paths.

Zusammenfassung

Eine wichtige Voraussetzung für die Realisierung von cyber-physischen Systemen (CPS) in kritischen Infrastrukturen wie Stromnetzen ist die Kommunikationszuverlässigkeit. Die Zuverlässigkeit wird üblicherweise im Hinblick auf die Nichtverfügbarkeit von Kommunikationsdiensten gemessen. In diesem Zusammenhang haben die für Smart Grids vorgeschlagenen Anwendungen Zuverlässigkeitsforderungen von 99-99,9999%. Um diese zu erreichen, sind die meisten Energieversorger auf dedizierte Netze und/oder Mietleitungen angewiesen. Die Alternative für solche Lösungen ist das Internet, das ein globales, kosteneffektives Netzwerk für CPS darstellt, die große geographische Gebiete umfassen. Leider ist die Zuverlässigkeit des heutigen Internets unzureichend und variiert im Laufe der Zeit. Ein weit verbreiteter Ansatz in anderen Bereichen zur Verbesserung der Zuverlässigkeit verwendet Redundanz in Bezug auf Kommunikationspfade und übertragene Daten. Dies erfordert jedoch Kenntnisse über die Topologie des Netzwerks, um die physische Trennung der verwendeten Pfade sicherzustellen, was im Falle des Internets schwierig ist. Selbst mit diesem Wissen sind die meisten der verfügbaren Multipath (MP) - Kommunikationsprotokolle durchsatzorientiert oder für dedizierte Netzwerke entwickelt worden und können daher nicht direkt verwendet werden. Dennoch wird erwartet, dass MP-Kommunikation die Kommunikationszuverlässigkeit des Internets verbessern wird.

In dieser Dissertation werden Datenduplikation und dynamische MP-Auswahl zur Laufzeit bei Verwendung mehrerer end-to-end-Pfade (e2e) vorgeschlagen, um die Kommunikationszuverlässigkeit für internet-basierte CPS zu verbessern. In dieser Hinsicht wird das Problem der Pfadauswahl als ein Optimierungsproblem formuliert, um die minimale Anzahl von e2e-Pfaden auszuwählen und die redundanten Daten durch die erforderliche Zuverlässigkeit zu begrenzen. Die e2e-Pfade werden unter Verwendung verschiedener Internetdienstanbieter und MP-Kommunikationsprotokolle realisiert. Darüber hinaus wurden reale Messungen zur Untersuchung der Zuverlässigkeits-gewinne der MP-Kommunikation im Internet durchgeführt. Die erhaltenen Ergebnisse beweisen die Existenz von e2e-Pfaden, die völlig unterschiedliche Netzwerke durchlaufen und folglich wahrscheinlich physisch getrennt sind. Sie zeigten auch, dass die gleichzeitige Nichtverfügbarkeit von verschiedenen Kombinationen aus 2 und 3 Pfaden 0% betrug. Diese Ergebnisse motivierten den Ansatz des Reliable Communication for Cyber-Physical Systems (RC4CPS) vorzuschlagen. RC4CPS ist ein e2e-Ansatz, der die inhärente Redundanz des Internets und das Konzept der MP-Transportprotokolle nutzt, um die Zuverlässigkeit zu verbessern. Es bietet eine Online-Überwachung und eine dynamische MP-Auswahl, die die Pfad-diversität und die Wahrscheinlichkeit der Nichtverfügbarkeit berücksichtigt, um die Zuverlässigkeits-gewinne zu maximieren. RC4CPS wurde in MATLAB für erste Auswertungen und dann unter Verwendung des iPRP-MP-Transportprotokolls (Parallel

Redundancy Protocol for IP-Networks) implementiert. Das resultierende Protokoll, das als iPRP-RC4CPS bezeichnet wird, enthält die RC4CPS-Funktionen und erweitert die ursprüngliche iPRP-Implementierung, die für dedizierte WAN-Netzwerke entwickelt wurde, um den Einsatz im Internet zu unterstützen. Die Auswertungsergebnisse, die unter Verwendung beider Implementierungen von RC4CPS im Internet durchgeführt wurden, zeigen die Fähigkeit von iPRP-RC4CPS, 0% Nichtverfügbarkeit zu erreichen, während die minimale Anzahl von e2e Pfaden ausgewählt wurde.

Contents

1 Introduction

Technological advances in communication and computation fields in the last few decades provided the possibility to develop very tiny devices with sensing, actuating, computation, and communication capabilities. Such combination created intelligent devices that are able not only to monitor but also to interact with the surrounding environment. More and more intelligent devices nowadays are being connected to the different types of communication networks, which enrich the linkage between the physical world and the cyber world. This integration between computation and physical processes by means of a communication network is usually referred to as a cyber-physical system (CPS) [1]. Applications of such systems cover a wide range of domains that include healthcare, transportation, energy and water infrastructures, industrial automation, environment monitoring, and smart buildings.

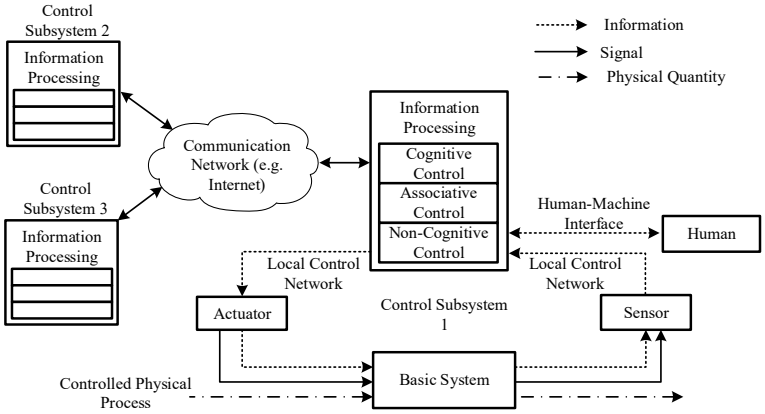


Figure 1.1 General Architecture of CPSs [2].

A CPS, as shown in Figure 1.1, consists of one or more interconnected autonomous subsystems or units. A CPS unit generally consists of the following entities: an information processing entity to monitor and control the physical process, sensing and actuating entities to interact with the physical process, and the physical process to be controlled. The information processing entity is expected to be able to change the nature of the rigid and reactive connection between the sensing and actuating entities provided in older embedded systems [2]. Such information processing unit can be described by a three-layer model with non-cognitive control, associative control, and cognitive control. Each of these layers has different functions. The non-cognitive control layer is responsible for the continuous control of, for example, active chassis of a car. The associative control layer is responsible for, but not limited to, conditional and stimula-triggered control decisions. Lastly, the cognitive control layer will include functions related to artificial intelligence (e.g. self-optimization). Another distinctive feature of CPSs is networking that is needed not only at local levels within each unit, but also at higher levels between the units.

CPSs were envisioned to be networked in a way that the services of each unit are visible to the other units of the system. This allows information exchange not only at local levels within the units of the CPS but also between the units. With this regard, local control networks are usually used to realize communications within the CPS units due to the real-time (RT) communication requirements of the technical processes under control. In contrast, the RT requirements between the units of the CPS are usually less stringent and, consequently, other types of networks including the Internet might be used to realize the communications. A key requirement on communication infrastructures for CPSs is reliability.

When considering CPSs, the reliability of the different types of communication networks that might be used within a CPS becomes a crucial issue. The IEEE Standard Computer Dictionary [3] defines reliability as "the ability of a system or component to perform its required functions under stated conditions for a specific period of time". An unreliable communication might cause not only financial costs and service disruptions but also human fatalities [4], [5]. In fact, one of the key requirements to realize CPSs in critical infrastructure is to have reliable communication networks [1], [6]. This is because control loops in CPSs might be closed over networks. If the network is unreliable, then degradation or destabilization of the control loops are expected. Even though that several CPSs have requirements on the network performance; it is unlikely that such requirements can be guaranteed in an unreliable network. This is mainly due to severe and frequent disruptions of communication service that occur in unreliable networks [7]. As a result, communication infrastructure of a CPS can be compared to the nervous system in humans that connects and coordinates the different body parts [8].

It is necessary to indicate here that the selected measure for reliability from the reliability theory [9] is availability. Availability is defined in [3] as "the ability of a system to be in a state to perform a required function at a given instant of time or at any instant of time within a given time interval; assuming that the external resources, if required, are provided". If the availability and reliability at a given time t are denoted as $A(t)$ and $R(t)$ correspondingly, then $A(t) = R(t)$ for a non-repairable system. For a repairable system, $A(t)$ is equal or greater than $R(t)$ [10]. By contrast, the unavailability for a repairable system denoted as $U(t)$ is equal or less than the unreliability denoted as $F(t)$. Consequently, the reliability requirements can be translated into availability/unavailability requirements, but not vice versa in the case of repairable systems. For example, a reliability requirement of 99% at time t can be translated into an availability requirement of at least 99% (i.e. $A(t) \geq 99\%$) or into an unavailability requirements of at most 1% at time t, in which case $U(t) = 1 - A(t)$. Some works in the literature (Section 3.1.3) considered availability as an attribute of reliability that is required to measure it. Other works indicated also that the availability and reliability terms are often used interchangeably and that the requirements on reliability are informally translated into requirements on unavailability. As will be indicated later in

Section 5.1, the use of unavailability is easier for the calculations. The use of unavailability is also motivated by the complex nature of the Internet where a large number of components contribute to the overall reliability of an end-to-end (e2e) path. By considering unavailability, the entire e2e path is considered as a single entity that is available or unavailable at time *t*. This agrees with the definition of unavailability given as the ratio of time a system is actually not functioning to the total time it is required to function [43]. If failures are assumed to occur for a noticeable time, then a very low unavailability indicates a very low number of failures. In other words, a very low unavailability indicates a very high reliability. Hence, reliability requirements presented in this work are implicitly translated into requirements of maximum unavailability and the main goal of improving reliability is achieved by decreasing unavailability.

In this dissertation, CPSs that can largely benefit from using the Internet to connect their different units are considered. Such CPSs are usually referred to as Internet-based CPSs. An example of such systems is smart grid. With this regard, the main focus of this work is on improving the communication reliability when using the Internet to connect the units of CPSs. As mentioned above, the provided reliability is measured in terms of communication service unavailability. The use of the unavailability measure does not only facilitate the evaluation and modeling of Internet paths but also reduces the complexity of the proposed solution. More specifically, this dissertation proposes an e2e approach during the course of the dissertation to provide reliable communication for Internet-based CPSs.

1.1 Motivation

When it comes to CPSs where the units are distributed over large geographical area as in the case of smart grids, Internet is a cost effective solution with very attractive features. Among these features, a few are described here. The Internet offers almost global connectivity with a wide range of wired and wireless access technologies that suit the requirements of different geographical locations. The low cost of communications using the Internet is yet another important feature. Also, the Internet offers high flexibility to do changes later with regard to, for instance, the used access physical medium or the needed data rate without entailing high costs (e.g. by choosing another service provider). However, a key requirement of CPSs on the different communication networks is to provide high communication reliability.

The reliability of today's commercial communication networks and the Internet in general was considered to be inadequate to support many CPSs. Particularly, measurements indicate that unavailability of Internet paths is often above 1% [11], [12] while many CPSs require a lower value [1], [13], [8]. As a result, existing CPSs usually use dedicated networks or leased lines in order to provide the desired level of communication reliability. Therefore, a solution to improve the communication

reliability of the Internet is needed to enable its utilization for CPS in critical infrastructures.

One of the main challenges that need to be considered with this regard is the random nature of Internet behavior. This is mainly because Internet is a publicly shared network that provides best effort type of service. It has non-transparent infrastructure and include vast amount of networks and middleboxes [14] that cannot be controlled by its end users. The heterogeneity of networks to access the Internet and connect its different parts and the high variety of their characteristics contribute significantly to reducing Internet reliability. This is mostly stemming from the nature of each of the individual networks, each of which has different reliability limitations. Such limitations are attributed to failures, oversubscriptions, and other anomalies in the individual networks. Therefore, the Internet inherently has unpredictable reliability levels and does not provide the needed service grantees for reliable operation of CPSs [15]. From the above, the proposed solution must utilize the Internet as it's and rely only on the end-systems connected to it. Hence, the solution is expected to take the form of a communication protocol for the Internet. In general, end-systems connected to the Internet use the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack. In this stack, only the transport and application layers are able to provide e2e services between end-systems. By contrast, the lower layers are responsible about the communication between the network components along the path between end-systems.

By considering other domains such as public telephone networks and control networks for power substation automation, high communication reliability is also demanded. The mostly adopted approach mainly utilizes redundancy in terms of links, components, or even complete cloned networks and concurrent transmission of duplicated data [16]–[19]. This achieves multipath (MP) communication and significantly improves reliability. However, the utilization of such approaches usually requires control over the network topology, which is difficult in the case of Internet. Nevertheless, MP communication is still expected to improve the communication reliability of the Internet. Fortunately, a few MP communication protocols such as iPRP (the Parallel Redundancy Protocol for IP Networks) [17] and MPTCP (MultiPath TCP) [20] have been recently developed in the transport and application layers of the TCP/IP stack. As shown in Figure 1.2, these mentioned MP protocols, for example, allow end-systems to connect to more than one IP network and support the concurrent transmission of data over the available network interfaces. At this point, the complexity of the Internet's infrastructure does not necessarily need to be a disadvantage. Due to today's large number of Internet service providers (ISPs) that operate autonomously and each run and extend its own physical networks, a vast number of independent Internet peers have been formed. This resulted in multiple e2e paths in between them. The utilization of this feature with appropriate MP communication protocols to create network redundancy seems a potential solution.

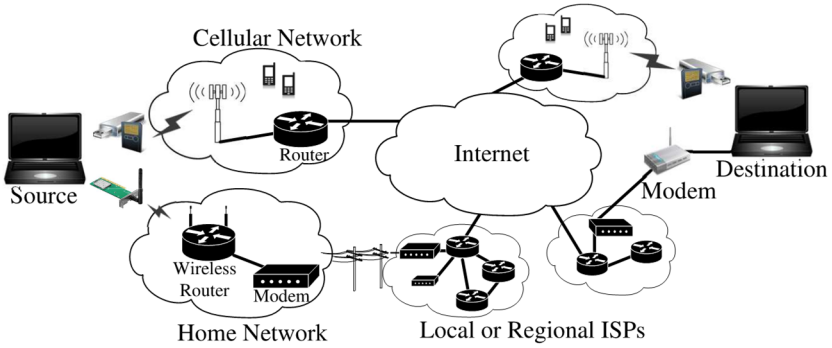


Figure 1.2 Concurrent utilization of different access ISPs when using MP communication protocols.

Unfortunately, all proposed MP communication protocols for IP networks are throughput-oriented, proposed for dedicated networks with controlled topology, or do not consider the required reliability for CPSs and maintaining it in the Internet. In addition, most of these protocols are incompatible with middleboxes.

Based on the above-mentioned observations, the main motivation of this dissertation is to develop an e2e solution that provide adequate reliability for Internet-based CPSs while considering the technical challenges in today's Internet.

1.2 Structure of the Thesis

This Dissertation is organized as follows:

In *Chapter 2*, I provide the research description and start with the problem description. The research objectives and questions are introduced after that. Lastly, the contributions of the dissertation are highlighted.

In *Chapter 3*, the state of the art in a number of aspects is analyzed. More specifically, I look to the existing approach to improve communication reliability of the Internet. The approach proposed in this dissertation to improve the communication reliability of the Internet utilizes the concept of MP communication protocols. Therefore, I present the literature review of MP communication protocols proposed for IP networks in this chapter. The last section of this chapter indicates the research gap for the considered research problem.

Chapter 4 provides an overview for a number of concepts considered in this dissertation. The overview considers data communication network in general, communication reliability, impacts of communication network deficiencies on reliability, CPSs and the different types of communication networks used in such systems, and smart grid applications.

Chapter 5 presents the proposed approach with the name "Reliable Multipath Communication for Internet-based CPSs (RC4CPS)." In the first section, the system model considered is discussed. Then, the optimization formulation for MP selection to

provide MP communication is provided. After that, the architecture of the approach is described. In addition, the online procedures performed by the MP Selection component of RC4CPS for path selection are detailed. This includes the selection of the primary and backup subsets of paths and when the switching from one subset to another is done.

In *Chapter 6*, the characteristics of Internet paths in terms of diversity and unavailability between multihomed end-systems are investigated. In this chapter, detailed description of the selected end-systems, locations, ISPs, and the evaluation setups are described. In addition, the evaluation results are provided.

In *Chapter 7*, the mechanisms used in the Monitoring & Estimation (M&E) component of RC4CPS to carry out its tasks are described. These tasks include the monitoring of e2e paths and the estimation of the MP selection attributes described in *Chapter 5*. The possible models to be used by this component to model the different e2e paths are also described.

Chapter 8 describes the implementation consideration for RC4CPS. Most importantly, the chapter analyzes the already existing MP communication protocols at the application and transport layers of the Open System Interconnection (OSI) model with regard to these considerations.

In *Chapter 9*, the importance of confidence interval (CI) in evaluating the goodness of the estimated reliability benefits for MP communication is highlighted. The parametric and non-parametric approaches for CI estimation are also presented. The estimated CI based on sample data from Chapter 6 is provided after that.

In *Chapter 10*, an implementation of RC4CPS using MATLAB [21] is detailed. The implementation targeted providing an evaluation platform of RC4CPS and the adopted mechanisms in it. The evaluation results of the implementation using multihomed end-systems and real-world Internet e2e paths are also presented.

Chapter 11 introduces the MP transport protocol selected for the implementation of RC4CPS to provide a real-world, ready-to-use implementation of the approach. The details of integrating RC4CPS approach in the existing protocol implementation are also provided. After that, the results of the protocol evaluations done in a lab environment and in the Internet are presented.

Lastly in *Chapter 12*, I conclude the dissertation. Additionally, an outlook for future work is provided.

2 Research Description

This chapter aims at describing the research problem considered in this dissertation. Then the research objectives and the resulting research questions are introduced. Lastly, the author's contributions to the topic are indicated.

2.1 Problem Description

As mentioned in Chapter 1, CPSs are/will be deployed in critical infrastructure such as energy and water infrastructures. In such domains, the expected reliability of CPSs and the used communication networks within them is very high [1], [8], [22]. Unreliable communication networks within a CPS might result in destabilization of the control loops closed using them. This in turn might result in financial costs and service disruptions [4], [5]. A challenge in this area is to provide reliable communication for CPSs that span large geographical areas such as smart grids. In this case, the use of Internet to connect the geographically distributed components of the CPS seems a very cost effective solution. However, such systems usually use dedicated networks or leased lines rather than using the Internet due to its inadequate reliability. Different measurements with this regard show that the unavailability of Internet e2e paths is often higher than 1%. On the other hand, the unavailability requirements of different smart grid applications, for example, are in the range 0.0001-1% [8]. With this gap between the provided and required unavailabilities, the utilization of the Internet is not possible.

The challenge of improving the Internet reliability is a long standing one, not only in the domain of CPSs. This is attributed to several reasons including the best effort type of service and the complex infrastructure of the Internet. Even trying to improve reliability through new TCP/IP protocols residing at the end-systems is very complicated. This is because several technical challenges in today's Internet hinder this. For example, the various types of middleboxes deployed in the Internet today are tailored to existing protocols. More specifically, some middleboxes such as firewalls work based on a white list approach. Only explicitly allowed flows with known contexts and structures of packets are able to pass. These are usually the flows carried out by TCP and UDP (the User Datagram Protocol) or protocols based on them.

As mentioned previously, redundancy in terms of communication paths and transmitted data is widely used to improve reliability of communication networks in the different domains. Nevertheless, the expected improvement in communication reliability using MP communication cannot be easily estimated in case of the Internet. This is attributed to the nature of the Internet infrastructure which cannot be controlled by the end users and is also non-transparent for them. Moreover, Internet infrastructure is continuously changing physically (by adding new links and devices or due to new peering agreements between the different ISPs) or logically (by changing the routing rules). As a result, two

e2e paths in the Internet might share some links and the concurrent utilizations of them might not improve communication reliability significantly. Even if it is assumed that MP communication can support the reliability requirements for Internet-based CPSs, existing MP protocols do not give end users control on paths selection. For example, MPTCP establishes a full-mesh of e2e paths between available source and destination network interfaces (or IPs). However, using all e2e paths between two end-systems for data replication is not desired. The unnecessary use of all e2e paths between two multihomed end-systems might result in waste of network resources, reduces approach scalability, increase overhead to handle duplicated packets at the receiver, and might also increase cost.

Solutions that target improving Internet communication reliability for CPSs are almost non-existent. In additions, the proposed solutions to improve communication reliability for CPSs fall short of one or more of the following criteria: (i) they do not consider the reliability requirements of CPSs when establishing the communication, (ii) they are proposed for local control networks or dedicated wide area networks (WANs), (iii) they are not scalable as they require additional equipment and/or cooperation with network equipment between end-systems, or (iv) they are not fault-tolerant against service disruption occurring at the access ISP (use single-homed nodes).

In summary, CPSs were envisioned to have IP capability and to use the Internet as their future communication network [2], [23]. However, the vision has not yet become a reality, at least not in a widely accepted and broad sense. This is mainly because of this gap between the offered and demanded reliability. As a result, the potential and advantages of CPSs in the different aspects and areas of human life cannot be fully utilized.

2.2 Research Objectives and Questions

As indicated in Section 1, this dissertation focuses on CPSs that span large geographical areas where the use of Internet is demanded. Due to the wide spectrum of CPS applications, I consider only those in the domain of smart grids. Power grids seem one of the earlier CPSs domains that witnessed a lot of research due to their expected economic and environmental impacts. Moreover, standards such as IEEE C37.118 [24] and IEC 61850 [25] have been developed and consider the communication requirements of these applications. The objectives of the research can be extended to other applications that can benefit from the Internet.

The ultimate goal of this dissertation is to find an approach to provide adequate reliability for CPSs when using the Internet. Such approach should rely on end-systems only and do not require any support from the networks. In this sense, the approach should deal with the Internet as a black box. This is due to the complex nature of the Internet infrastructure in which the end users have no control over it. By relying on end-

systems only, the scalability is increased and the approach can be widely deployed in the future. In addition, the approach should consider not only the reliability requirement of the CPSs, but also the technical challenges imposed by the Internet.

In order to meet the above-mentioned requirements, it is clear that a communication scheme residing at the end-systems and realized through a communication protocol is needed. This protocol must be placed at the transport or application layer of the TCP/IP stack to manage CPS e2e communications. However, I have indicated in Section 2.1 that the wide use of middleboxes in today's Internet hinder the deployment of new communication protocols. Hence, it is necessary to consider only existing protocols that are compatible with middleboxes.

In recent years, a number of MP communication protocols were proposed to utilize the inherent redundancy of the Internet. Some of these protocols consider middleboxes and their interactions. The concept of such protocols represents a potential solution to the problem considered here. More specifically, these protocols enable end-systems to establish more than one e2e path through (possibly) different networks to transfer data. With such concept, redundancy, which is a widely adopted mean to improve reliability, can be provided through MP communication. Nevertheless, all these protocols have one or more of the following issues: (i) they target maximizing throughput rather than reliability, (ii) they were proposed for dedicated IP networks with controlled infrastructure to control the established paths, or (iii) they provide passive interaction with the application such that the desired reliability has no influence on the e2e interaction and the used paths.

In a nutshell, I propose an e2e approach for dynamic MP selection during runtime when utilizing different e2e paths to cope with the varying nature of Internet and to provide the required reliability for CPSs. As mentioned previously, these e2e paths can be realized using different ISPs and a MP communication protocol. The MP selection targets limiting the redundant data by the needed reliability and considers paths diversity and unavailability. In this context, the main objective of the dissertation can be divided into three sub-objectives.

Sub-objective 1: To evaluate the reliability benefits of MP communication over the Internet

The Internet infrastructure is non-transparent and consists of a vast amount of nodes, links and networks. In such complex network of networks, e2e paths established using MP protocols and different ISPs might overlap or at least traverse the same networks. As a result, any service disruptions over the shared links and/or networks will impact all e2e paths that traverse them. In such situation, the reliability benefits of MP communication diminish due to the single point of failure.

In this first sub-objective, I target evaluating the diversity and unavailability of different e2e paths. In addition, I evaluate the reduction in communication service unavailability when multiple paths between communicating end-systems are considered.

Sub-objective 2: To develop a concept for a reliable MP communication approach for Internet-based CPSs

It might seem straightforward to duplicate data packets over all possible e2e paths to improve reliability. However, packet duplication over all available e2e paths is not desired. First, it might not be cost effective if carriers charge per data volume carried. Second, for the approach to be scalable and to be widely deployed in the future, the redundant data needed for improving reliability should be minimized for each pair of communicating parties. This avoids wasting network resources and does not result in networks congestions when there is a large number of devices. Third, it increases overhead at destination to handle duplicated packets.

In this second sub-objective, I target providing a concept along with its architecture to achieve reliable MP communication. This concept should consider minimizing the number of e2e paths utilized while providing adequate communication reliability for the communicating CPS units. This minimization of the used e2e paths should be done during runtime and should consider a number of attributes. These include: (i) the achieved reliability measured in terms of communication service unavailability. (ii) The diversity between the e2e paths based on e2e measurements. This needs to be done without cooperation with the network components or knowledge about the underlying topology. (iii) The characteristics of the e2e paths and the ability to predict their future behavior.

Sub-objective 3: To provide a feasible and easy-to-deploy real-world implementation of the approach

As highlighted in Section 2.1, new protocols for the TCP/IP stack need to be based on the legacy protocols, TCP and UDP. Therefore, the implementation of my approach should be done using existing protocols rather than proposing a new one. In this third sub-objective, I target first evaluating existing MP protocols and their deployability in today's Internet. Then, the adaptability of these protocols to realize the approach to be developed according to sub-objective 2 is evaluated. Lastly, I target providing a real-world implementation of the approach in the form of a modified MP communication protocol.

Research questions

The main research question (RQ) of this dissertation, based on the previously mentioned objectives, is the following: **How to provide reliable MP communication for CPSs using the Internet.** This main RQ can be splitted into two detailed RQs.

RQ1. **How MP communication using different e2e paths could improve reliability?**

RQ2. **How to select the minimum set of e2e paths to provide the required communication reliability?**

RQ1 considers investigating the diversity and unavailability of multiple e2e paths when using different pairs of access ISPs to connect two end-systems. It investigates the existence of disjoint path and the achieved reductions in communication service unavailability when multiple paths are considered concurrently.

RQ2 considers the development of a concept that enables the selection of a subset of all e2e paths during runtime based on a number of attributes to satisfy the application desired reliability. This requires determining which attributes are needed and the mathematical formulation for the selection. Furthermore, the architectural description of the concept and the interaction between the different components and their corresponding attributes are also considered.

2.3 Contributions

The publications resulted from this dissertation are [26]–[33]. The most important of these are [26]–[28], [33]. These later publications address directly the stated research questions. They investigated the potential of MP communication over the Internet to support the reliability requirements of CPSs. They also present the proposed approach called "**Reliable Multipath Communication for Internet-based CPSs (RC4CPS)**" and the evaluation results for its implementations in MATLAB [21] and using the iPRP MP transport protocol [17]. The later implementation offers an easy-to-deploy solution as it is a transport protocol based on UDP. Therefore, no modifications are needed at the application layer and there are no compatibility issues with middleboxes. In addition, the evaluation of existing MP communication protocols with regard to the implementation requirements of RC4CPS is also carried out in one of them.

The contributions that were developed during this work can be stated as follows:

1. Conduct an extended real-world measurement considering a large number of Internet e2e paths established using different pairs of ISPs. The results indicate the existence of e2e paths that are traversing completely different networks (likely to be disjoint). In addition, the results show that MP communication can support the reliability requirements of 99.9999% of some CPSs such as smart grids.

2. Develop an e2e approach to provide reliable communication within CPSs using the Internet called RC4CPS**.** It provides online monitoring with online and dynamic MP selection in order to fulfill the application specific reliability requirement. The MP selection in RC4CPS considers also e2e paths diversity and unavailability prediction to cope with the varying nature of Internet paths**.**

3. A real-world implementation of RC4CPS using an existing MP transport protocol with simple deployment procedures and no need of additional component between

communicating end-systems. The resulting protocol, called iPRP-RC4CPS, requires also no cooperation/modification with/on the network components. The conducted evaluations show that iPRP-RC4CPS can support CPSs such as smart grids with reliability requirements of 99.9999%.

3 State of the Art

In this chapter, I first present the literature review of the key concepts and terms along with the relationships between them. In the later sections, the related work regarding improving the communication reliability for CPSs as well as for the Internet in general is presented. Lastly, the research gap regarding the topic is highlighted.

3.1 Introduction to the Main Concepts

This section introduces two key terms, namely CPSs and reliability. With this regard, the literature discussing the importance of reliable communication and the reliability requirements for CPSs is presented. The relationship between CPSs and similar networked systems is also described in this section. In addition, the relationship between the terms *reliability* and *availability* is discussed. The related work to be presented in this section is summarized in Table 3.1.

3.1.1 CPSs and Associated Literature to Reliability Analysis and Requirements

A CPS, as defined by Lee [1], is an integration of computation with physical processes. According to the vision introduced by Lee, networking is the main feature that characterizes this new paradigm of control systems. In CPSs, computation components monitor, control, and coordinate physical and engineered systems through communication networks. One of the key requirements for CPSs, first indicated by Lee, is communication reliability. This was also confirmed by several works in the literature (e.g. [8] [22] [23]). Moreover, the need for higher communication reliability has also been considered in the development of the 5th Generation (5G) of wireless access technologies. This is mainly to enable new use cases in the domain of mission-critical Machine-Type Communication (MTC) with a requirement of low unavailability [34]. Hauser et al. [4] considered the requirements on the next generation communication networks for power grids. The authors indicated first that the current communication technologies and infrastructures used in power grids are very old (several decades old). Consequently, stability problems cannot be reported fast enough to prevent service disruptions. The authors also highlighted the need for reliable communication and the benefits of using IP networks for power grids. These benefits include the support for a large number of communicating components and multicasting capabilities. In addition, the use of IP networks along with middleware techniques would allow comprehensive solutions for e2e communications. The impact of cyber part reliability including the communication network on the overall CPS reliability in the domain of smart grids has been presented in [35], [36].

Table 3.1 Summery of related work presented in Section 3.1.

Related Work	Topic
[1]	CPSs definition
[1], [8], [22], [23], [34], [4], [35], and [36]	Importance of communication reliability for CPSs
[6], [4], [8], [13], [37], and [38]	Reliability requirements for CPSs
[39] and [40]	CPSs relationship to other networked systems.
[3], [41], [41], and [42]	Reliability definition
[3] and [56]	Availability definition
[44], [45], [46], [9], [10], [47], and [13]	Relationship between reliability and availability.
[7] and [47]	Issues deteriorating Internet reliability
[48], [11], and [49]–[51]	Measurements regarding reliability of Internet paths
[52]	Measurements regarding the MP nature of the core parts of the Internet
[11], and [53]–[55]	Measurements regarding the reliability benefits of multi-homing

In order to provide reliable communication to CPSs, it is very important to understand their communication requirements. The scope with this regard is limited to smart grids domain. A number of smart grid applications along with their traffic characteristics and communication needs were presented in [47], [8], and [13]. A more detailed survey of the communication requirements of applications in smart grids was conducted in [37]. Analysis of the capability of WANs to support IEC 61850 based communications were carried out in [38]. The authors indicated in the case of inter-substation communication the applicability of time-critical functions for protection. However, fiber optic links connecting substation routers and carried over the power lines were assumed as the WAN links rather than using public networks such as the Internet.

3.1.2 Relationship to Other Networked Systems

The concept of combining computing and physical processes has been already considered in engineered systems. Such systems have existed since a few decades and are usually called “embedded systems”. Examples of embedded systems include home appliances, aircraft control systems, and automotive electronics. The main difference between embedded systems and CPSs is that embedded systems represent mostly black boxes. They do not show their computing capability to the outside and no outside connectivity can alter their software behavior [39]. In a CPS, the units of the system are feature-rich, networked, and cooperate together using communication networks.

CPSs have also some similarities to other related networked systems such as machine-to-machine (M2M) networks and wireless sensor networks (WSN), Wan et al. [40] pointed out that CPSs represent a wider concept and can be considered as an evolution

of M2M and WSNs, but with decision-making capabilities and autonomous control. As presented in [40], these networked systems have the same main components with different proportions which are: information sources such as sensors, information sinks such as embedded computers, communication networks to carry information between sources and sinks, and finally applications and services that utilize such networked systems. In addition, the distinctive features for each of these systems were also discussed in [40]. M2M networks are more concerned with the communication between devices (which might involve human interaction) while CPSs focus, in addition to communication, on coordinating and optimizing the control functions of the CPS units and create intelligence in the networked system with decision making capability. WSNs are more concerned with collecting the data from the physical environment where the sensory unit has only the functionality of monitoring. By contrast, each subsystem or unit in CPSs must provide control capability besides monitoring.

3.1.3 Internet Communication Reliability

In the literature, similar definitions for reliability were provided. Reliability, as defined in the IEEE Standard Computer Dictionary [3], is "the ability of a system or component to perform its required functions under stated conditions for a specific period of time". Pradhan [41] defines reliability to be the conditional probability that the system will carry out its desired function successfully at time t given that it was operating successfully at time $t = 0$. In [42], [43], reliability was defined to be a measure of correct service continuity.

A related concept to reliability, which is often used interchangeably with it, is availability [44], [47]. In [3], availability is defined as "the ability of a system to be in a state to perform a required function at a given instant of time or at any instant of time within a given time interval; assuming that the external resources, if required, are provided". In [56], system availability was defined as the readiness for usage. The availability is calculated as the ratio of time a system is actually functioning to the total time it is required to function and it is usually expressed as a percentage [44]. Several works in the literature tried to clarify the relation between these two concepts. For example, the author in [44] indicate that a highly reliable system is not necessarily a highly available system. On the other hand, Al-Kuwaiti et al. [45] and McCabe [46] considered availability as an attribute of reliability that is required to measure it. More particularly, McCabe argued that an unavailable system does not fulfill the specified requirements before all else to be reliable. In this context, Al-Kuwaiti et al. [45] considered availability to be reliability evaluated at a certain instant. In [9], [10], two cases for the relationship between the availability function, denoted by $A(t)$, and the reliability function, denoted by $R(t)$, were differentiated at a given time t. For non-repairable systems $A(t) = R(t)$ and for repairable systems $A(t) \geq R(t)$. If the unavailability and unreliability functions denoted by $U(t)$ and $F(t)$ correspondingly are considered, then $U(t) = F(t)$ for non-repairable systems and $U(t) \leq F(t)$ for repairable systems.

Consequently, the requirements on reliability can be expressed as requirements on availability/unavailability, but not vice versa in the case of repairable systems. In this research, the interpretation of relationship between reliability and availability provided by McCabe and Al-Kuwaiti et al. is adopted due to its convienience. This also agrees with how the communication reliability requirements for many CPSs are specified. Namely, as the amount of time that the network will be unavailable in a year [47], [13]. As an example, 99.99% reliability means that the unavailability of the network will be less than one hour ($0.0001 * 365 * 24 * 60\ min$) in a one-year interval.

As mentioned in Section 1.1, the Internet is a network of networks where each has its own reliability limitations. With this regards, the unreliable nature of general purpose commercial networks (which are part of the Internet) was presented in [13]. It was indicated why most utility providers prefer to use their dedicated communication networks. The issues raised in [13], included the inadequate priority of service provided and the inability to reduce the communication service unavailability due intermittent congestions. In addition, the unreliability of telephone and data networks in the USA and the challenges to providing reliable communication were also discussed by Snow [7]. Among the mentioned factors that contribute to reducing network reliability are complexity caused by concentrated infrastructure, the variety of used technologies and provided services, business revenue plans, market competition (which necessitate fast market entry), and rapid technological advances. All of these factors caused several service outages and severe service disruptions for the considered networks. For example, tracked outages affecting more than 30000 service users for 30 minutes or more over a period of eight years were reported to occur 14 times every month approximately. In the examples provided in [7], outages were usually occurring after certain events such as procedural errors and software and hardware upgrades. This in turn might be attributed to non-highly qualified staff and difficulty to estimate all interactions in complex and large networks.

In the literature, a number of approaches were used to evaluate unavailability of e2e paths in the Internet. The approach used in [48] utilizes two data sets where one set consists of *Traceroute* data of measurements between different pairs of nodes and the other set consists of HTTP requests data collected from public Squid caching proxy to Web servers. The authors used a binary on/off model to describe service unavailability based on request-average unavailability (the fraction of requests in a data set that fail in accessing services). The *Traceroute* data set showed that the average unavailability of Internet paths is ranging from 0.7% to 1.9%.

TCP acknowledgment (ACK) probes were used by Gummadi et al. in [11] to characterize path failures in the Internet. Unlike TCP-based probes, UDP and the Internet Control Message Protocol (ICMP) probes cause more security alarms and might be dropped by routers and firewalls or assigned lower priorities. The authors indicate the complication arise when trying to distinguish between packet loss caused by

true path failure and that caused by congestion. Nevertheless, the authors considered the loss of four probe packets consisting of one regular probe packet and three consecutive failure detection probe packets as a path failure. The duration of the failure was defined to be the time elapsed between the send time of the first of four failed probe packets till the send time of the first probe in a sequence of ten successful ones. The study measured unavailability from geographically distributed vantage points and considered Internet paths to a number of broadband end-nodes as well as to popular Web servers. The measurements showed an average unavailability of 0.4% for paths to popular Web servers and of 5.6% for paths to broadband end-nodes respectively. It was also observed that only 22% of paths to servers and 12% of paths to broadband end-nodes were failure-free. Other measurements were carried out in [49]–[51] and show similar results with average unavailability of Internet path ranging between 1.5% and 3.3%.

From the above studies it is clear that unavailability of communication paths over the Internet is often higher than 1%. In addition and as indicated by Gummadi et al. and Dahlin et al. [48], stub networks connecting end-sytems contribute significantly in such unavailability. For example and as mentioned above, Interent paths to popular web servers show lower unavailability compaterd to paths to broadband nodes. This is attributed to the high reliability of the stub networks connecting these servers to the Internet and, in some cases, the use of multihoming.

In [57], a study that considered five different datasets of measurements regarding quality of Internet paths showed that 30-80% of the cases, alternate paths with significantly superior quality were available. The reliability benefits of multihoming and overlay networks were investigated in [11], [53]–[55]. Here, multihoming was considered at one side of the communicating parties. Also, the benefits of overlay networks were considered with regard to reaching certain destinations connecting to the Internet through a single access network. In this work, the source and the destination connect to the Internet through different access networks (both are multihomed). In [52], it was observed that the use of multiple IP addresses between end-systems over WANs provided multiple diverse paths.

3.2 Approaches for Improving Communication Reliability for CPSs

In the literature, several works considered communication reliability for CPSs. In this section, the efforts carried out with this regard are presented. In [58], preliminary wireless system architecture for CPSs targeting providing high-reliability communication was proposed. The architecture targets industrial control systems to replace wired connection between many sensors and actuators while providing comparable reliability. Li et al. [8] provided a detailed overview about communications in CPS and proposed the modeling of the CPS communication infrastructure as a hybrid system with discrete and continuous system states. In [59], the authors proposed a two

layer network architecture for large M2M networks to improve information dissemination reliability. This is accomplished by connecting the lower machine swarm layer consisting of many small M2M device clusters through interconnected data gateways in the upper layer which forms ultra-fast shortcuts between the lower layer clusters. The authors in [60] analyzed the reliability of the neighborhood-area network (NAN) for demand-side management (DSM) in smart grids. They proposed three redundancy design approaches. The approaches targeted minimizing the deployment and failure costs of wireless communications for DSM. A hybrid communication technology, combining power line carrier and ZIGBEE technologies, to improve communication reliability in electric vehicle charging systems was presented in [61]. A transport protocol for sensor networks with adaptable reliability that determine the amount of data to be reliably transmitted based on the estimated error from its omission was proposed in [62]. Another transport protocol for future packet-switched railway signaling systems that is based on MPTCP [20] was proposed in [63]. However, the improvement in availability using the proposed protocol was evaluated by only modeling the multiple e2e paths as a common parallel system. The work does not describe how the new protocol extends the original implementation of MPTCP and does not consider the diversity of the e2e paths. It is also not clear whether the path selection is done during runtime or how the availability of e2e paths is estimated. In [64], it was proposed to extend MPTCP to provide spatial and temporal redundancy for single-homed and multihomed nodes respectively. Nevertheless, the spatial redundancy was achieved by combining MPTCP and the Multiprotocol Label Switching (MPLS) technique which requires cooperation with network operators. The utilization of MP communication was one of the adopted means to improve reliability of automation networks for power substations [16]–[18], [65]. In these works, cloned local substation networks and/or dedicated wide area networks were considered. In [66], [67], MP selection algorithms for control networks were proposed. Park et al. [67] proposed a robust path selection algorithm for CPSs that exploits the MP diversity to give the set of paths satisfying a certain delay bound. In addition, the selection must adhere to a robustness level obtained based on the reliability violation probability. Lukasz et al. [66] proposed a possibly disjoint MP selection algorithm in MP industrial networks to improve reliability.

The above-mentioned solutions have a number of drawbacks including: (i) they were not developed for the Internet and require a controlled network topology; (ii) they propose new designs/architectures for the communication networks or protocols without considering the characteristics of the Internet, and (ii) the conducted evaluations are simulation-based and the achieved unavailability is either not indicated or inadequate (> 0.0001%).

The utilization of network coding to provide redundancy (by generating a larger number of packets than the received number of application messages) and to improve

communication reliability of wireless communications for telesurgical robot systems was proposed in [68]. Nevertheless, the use of network coding to improve communication reliability requires support for network coding by both end-systems. It also requires additional redundant data to be sent over the used paths or additional e2e paths (as in [69]). Most importantly, a minimum number of coded packets need to be received in order to retrieve all application messages. As a result, retransmissions might be necessary. Moreover, such approach might increase the time delay the application experience in order to receive the minimum number of packets to retrieve the original data.

Table 3.2 Summery of related work presented in Section 3.2.

Related Work	Approach
[58], [8], [59], [60], [61], and [62]	New designs/architectures for communication networks/protocols
[63], [64], [16]–[18], and [63]	Networks with path redundancy and/or MP communication protocols
[15], [66], and [67]	Overlay networks and/or MP selection
[68] and [69]	Network coding over single path or multiple paths

Disjoint MP selection in overlay networks to meet specific performance and reliability requirements of smart distribution grid was proposed in [15]. Here, additional intermediate nodes with overlay route monitoring and selection capabilities are needed. The end nodes as well as intermediate nodes are single-homed and unavailability events at the access networks might impact them (have a single point of failure). It is necessary to indicate at the end of this section that MP selection was considered only in these works [15], [66], [67].

A summary of the related work presented in this section is provided in Table 3.2.

3.3 Approaches for Improving Communication Reliability in Other Domains

In this section, the approaches proposed to improve Internet reliability without considering the reliability requirements of CPSs are presented. Forward error correction (FEC) and traffic allocation over multiple paths was proposed in [12]. Network coding was also suggested in [69] to improve MPTCP goodput in the case of diverse network conditions on the available subflow. The issues of such approaches based on network coding are indicated in Section 3.2.

A similar system model to the one considered in this dissertation was proposed in [70] to select a given number of e2e paths in which the reliability concerns are considered. More specifically, the authors proposed a correlation-aware MP selection to choose

paths with high diversity and enhance the utilization of network resources. However, it is not clear how their proposed approach deals with middleboxes. It was indicated in [71] that if a middlebox detected noncontiguous sequence numbers for the transport protocol, it drop/block the corresponding packets/connection. The authors indicated also some of the approach drawbacks, namely: (i) the MP selection is done before the data transmission, (ii) there is a need to have a dynamic MP selection to cope with varying network/routing environment (how to update the set of selected paths according to the current network conditions, and (iii) how to reduce the cost of probing to update the information about available paths and reflect their current conditions. With regard to issue number iii, it was indicated that using passive probing (obtaining path information from the actual data sent) is one possible solution. Nevertheless, an accurate approach to obtain MP characteristics and the correlation characteristics is needed. Lastly, the approach does not apply a packet discard mechanism at the destination. This will increase the processing overhead as each arrived packet copy need to be processed at the destination. Unlike the formulation in [70], this dissertation targets providing dynamic online selection of the minimum set of e2e rather than a specific set and considers unavailability beside diversity in the selection process. Moreover, all proposed approaches in this section do not achieve 0.0001% unavailability.

Table 3.3 Summery of related work presented in Section 3.3.

Related Work	Approach
[12] and [69]	FEC over multiple paths
[70]	Diversity-aware MP selection

A future trend that is also expected to enable the realization of Internet-based CPSs is the Tactile Internet [72] that is characterized by having very low latency and high reliability, security, and low unavailability. With such characteristics, the Tactile Internet will support not only existing applications but also new ones in a wide range of domains including industrial automation, healthcare, transportation systems, and gaming. Nevertheless, Tactile Internet imposes very stringent requirements on the communication infrastructure that might not be supported by existing technologies and demand the development of capable ones. Even with the existence of such technologies, single-homed end-systems are still expected to be impacted by unavailability events occurring at the access networks.

3.4 MP Communication Protocols

As mentioned in Section 2.2, I have proposed dynamic online MP selection when using MP communication to improve communication reliability for Internet-based CPSs. This is expected to be realized by using a proper MP communication protocol in the TCP/IP stack and different ISPs. The existing MP protocols cannot be utilized directly to implement RC4CPS or to provide reliable communication for Internet-based CPSs in general. However, deploying new protocols is also very difficult due to the technical

challenges of today's Internet. These technical challenges also hinder the utilization of many of the already proposed protocols. Therefore, RC4CPS will be implemented using one of the existing MP protocols. Hence, I provide the related work with regard to MP communication protocols in this section. A summary of the protocols presented in this section along with their main features is provided in Table 8.1.

To the best of my knowledge, Maxemchuck [10] is the first who pursued the idea of transmitting data over multiple paths between two end-systems in 1975. MP communication has also been proposed in the different layers of the OSI model. However, approaches in the network layer (e.g. [11]) do not provide end-systems with path statistics about time delay, packet loss, etc. They also do not take issues such as path congestion or packet reordering into account. Consequently, MP selection is difficult at this layer. Moreover, approaches at this layer and lower ones require either modification to physical infrastructure or cooperation with network components due to the point-to-point communication nature (e.g. the Link Aggregation Control Protocol (LACP) [12]). As a result, I will only consider MP protocols at the transport layer or higher.

The recent surveys and works in [73]–[76] list most MP protocols that have been proposed over the last decade. Recent MP protocols that were not included in these surveys were also taken into account. Not all the protocols in the surveys were considered in this section for the following reasons. First, some protocols allow only one end-system to be multihomed, but the interest in this work lies in protocols that allow both communicating parties to be multihomed. Second, some of the protocols propose only new packet scheduling or congestion control (CC) algorithms for existing ones. Therefore, evaluating the original protocols is adequate for this work.

In the transport layer, Multipath TCP (MPTCP) is one of the current and most sophisticated MP concepts. It is based on the legacy TCP protocol and is fully backwards compatible with it. Beside the connection-oriented services of TCP, MPTCP binds connections to multiple IP addresses and allows the establishment of multiple data subflows over available e2e paths. Several extensions were proposed for MPTCP. Network Coding Based MPTCP (NC-MPTCP) utilizes part of the available subflows to send redundant data to compensate for packets that are timed out or lost. Systematic Coding MPTCP (SC-MPTCP) and Fountain-Code-Based MPTCP (FMTCP) both attempt to reduce the overhead and encoding/decoding delays. Zhou et al. suggested the Congestion Window Adaptation MPTCP (CWAMPTCP) to address delay heterogeneity of e2e paths and improve MPTCP goodput. Similarly, MPTCP Slow Path Adaptation (MPTCP-SPA) tries to improve goodput by suspending bad paths. QoS-oriented MPTCP (QoS-MPTCP) offers out-of-order packet delivery and prioritizes information of most significance for real time applications. OpenFlow-MPTCP combines MPTCP and OpenFlow [17] to limit the used paths to a subset of disjoint paths only, however, OpenFlow enabled routers and switches are required. Augmented MPTCP (AMPTCP)

seeks to accomplish the same goal through the Locator/Identifier Separation Protocol (LISP) [18] on the network layer. MPCubic implements a modified CC algorithm into MPTCP to enhance bandwidth usage in networks with high bandwidth-delay products.

Another MP protocol at the transport layer is the Stream Control Transmission Protocol (SCTP). It was considered to be the future successor of TCP and UDP and should fix most of their deficits. It provides all of TCP's connection-oriented functions as well as UDP-like unordered delivery. Several extensions of SCTP were proposed to fix flaws and to add more functionality. Noteworthy examples include: Concurrent Multipath Transfer SCTP (CMT-SCTP) that provides concurrent data transmission using SCTP. Resource Poolingenabled CMT-SCTP (CMT/RP-SCTP) that targets providing fairness towards other SCTP or TCP flows. Dynamic Address Reconfiguration SCTP (DAR-SCTP) that allows dynamically adding or removing IP addresses to connections. Forward Prediction Scheduling SCTP (FPS-SCTP) that reduces the number of packets arriving out-of-order by distributing the data packets based on their estimated arrival time.

MPTCP and SCTP are not the only MP transport protocols. Parallel TCP (pTCP) is a wrapper around a modified TCP that stripes data over multiple TCP flows (called TCP-v pipes) with a shared send buffer and individual CC and loss recovery for each TCP-v pipe. Concurrent TCP (cTCP) uses a single congestion window and sender buffer along with a Credit-Weighted Round-Robin scheduler to split data over multiple cTCP subflows. Multipath TCP (M-TCP) focuses on increasing reliability in lossy wireless networks by using a modified Dynamic Source Routing (DSR) to allow MP routing and packet duplication. However, no packet duplicate discard mechanism is adopted at the receiver, resulting in performance degradation. Multipath Transmission Control Protocol (M/TCP) establishes different subflows in a connection using TCP options. It uses duplicate transmission over more than one path and also duplicated acknowledgments to provide fast retransmissions. Rate-based M/TCP (R-M/TCP) is an M/TCP extension and introduces a rate-based and loss-avoidance CC that utilizes estimations of queue length at bottleneck links. The detection of shared congestion between TCP subflows by utilizing Resilient Overlay Networks (RON) [19] was proposed in mTCP.

iPRP [7] is an extension of the Parallel Redundancy Protocol (PRP) [20] to support IP networks. iPRP is UDP-based and provides MP communication by establishing disjoint paths between multihomed systems in dedicated networks. In End-to-End Multipath Transfer (E2EMPT), data splitting is done in a Weighted Round-Robin (WRR) fashion, where the weight is based on available bandwidth, round-trip time (RTT), and packet loss. Reliable Multiplexing Transport Protocol (RMTP) uses packet pair probing to estimate the bandwidth of the paths and CC to determine the rate at which transmitted frames will not undergo queuing delays. Analogous to FMTCP protocol, Multi-Path

LOss-Tolerant (MPLOT) also uses packet coding to provide Forward Error Correction (FEC) and robust MP transmission on heterogeneous and lossy paths.

MP protocols at the application layer were also proposed. GridFTP is an extension for the standardized File Transfer Protocol (FTP). The protocol creates multiple TCP connections at the application layer for single-homed systems to increase throughput. MultiTCP utilizes multiple TCP connections for time-sensitive multimedia streams to minimize throughput fluctuation when congestion avoidance is invoked. Parallel Sockets (PSockets) and XFTP counter the problem of too small TCP window sizes that result in under-utilization of bandwidth in networks with large bandwidth-delay products. Nevertheless, all above-mentioned application layer protocols do not support multihomed hosts. Multipath RTP (MPRTP) is a MP communication model for the Real-Time Transport Protocol (RTP) [77] and is capable of adding or removing paths based on their quality. Lastly, the Multipath Transport System Based on Application-Level Relay (MPTS-AR) is a MP transport framework based on UDP and overlay networks with support for data duplication.

At the session layer, Deployable Bandwidth Aggregation System (DBAS) is a middleware that allows two multihomed devices to utilize all network interfaces without modifying applications or standard sockets. Green DBAS (G-DBAS) is an energy-aware extension to DBAS that balances the trade-off between throughput and power consumption, but supports only single path communication. To support MP communication in G-DBAS, an Optimal Energy Efficient Bandwidth Aggregation System (OPERETTA) was proposed. UDP-based Redundant Interconnection with Inexpensive Network (RI2N/UDP) and Multiple Network Interface Socket (MuniSocket) are both user level middleware that modify the UDP-socket to provide MP communication.

The majority of MP protocols considered above is throughput-oriented and focuses on exploiting multiple paths to increase throughput. Some of these protocols were proposed for dedicated networks with controlled network topology. Even though that all of the above protocols were proposed for the TCP/IP stack, the technical challenges of Internet might hinder their deployment (incompatibility with middleboxes). Hence, direct utilization of such protocols to improve reliability of internet communication is not feasible. More important, all of these MP protocols lack the connection to the desired reliability level by the application and the ability to adapt accordingly.

3.5 Research Gap in the Literature

Even though that reliable communication is a crucial requirement to realize CPSs, there was not much work with this regard under the scope of CPSs. More specifically, the work related to providing reliable communication within CPSs using the Internet and with consistency to the communication requirements imposed by such systems (see

Section 3.1.1) is almost extinct. Even the works discussed in Section 3.1.3 analyze the reliability gains in terms of path diversity when only one of the communicating parties is multihomed, but not when both are.

When considering the approaches described in Sections 3.2 and 3.3, most carried out evaluations are simulation based and the achieved unavailability levels do not support all smart grid applications (> 0.0001%). In addition, only [78] considers the use of Internet for CPSs. Nevertheless, the approach relies on overlay networks which necessitate the use of additional intermediate nodes. The placement of the intermediate nodes, the needed route monitoring and selection capabilities between them and the associated monitoring overhead, the single-homed nature of their connection to the Internet, and the extra e2e time delay that might be caused by them to process and forward packets limit the scalability of the approach and increase its deployment complexity. The other approaches presented in Section 3.2 were not developed for the Internet and do not consider its characteristics. Therefore, direct utilization of such approaches is not feasible. On the other hand, the approaches presented in Section 3.3 consider improving the reliability of Internet in general. Despite the fact that such approaches are expected to improve reliability, they do not consider the reliability requirements of CPSs explicitly. Therefore, it is not clear which CPS applications can be supported by such approaches. Moreover, their deployment in today's Internet, face a number of issues. The approach proposed in [15] shares the same challenges of overlay networks mentioned in Section 3.2. Beside the drawbacks indicated in Section 3.2 regarding the utilization of network coding, the approach in [12] resides at the network layer. Hence, the approach might have incompatibility issues with middleboxes [71]. The approach presented in [70] is the closest to my vision of how to provide reliable communication for CPSs using the Internet. Nevertheless, it has a number of drawbacks. Some of these were indicated by the authors such as the static MP selection before sending data and the probing overhead to monitor the paths. The approach does not also consider packet discarding at the destination.

The mostly adopted approach in the considered literature mainly utilizes redundancy in terms of links, components, paths, or even complete cloned networks. With this regard, MP communication protocols in the TCP/IP stack seem a good candidate to improve reliability. Unfortunately, all of these protocols cannot be utilized directly to improve reliability where each has one or more of the following matters: (i) it does not provide MP selection and creates a full-mesh between communicating IPs, (ii) it does not perform packet duplication and splits the data over available paths, (iii) it does not consider the presence of middleboxes and their interactions, (iv) it was proposed for local/dedicated networks and cannot be directly deployed in the Internet, or (v) it does not consider the reliability requirement of the CPS application (the required reliability level does not influence the protocol behavior).

To the best of my knowledge, there is no existing work in the literature that investigates the diversity and unavailability of e2e paths when both end-systems are multihomed to different ISPs. More specifically, the existing works investigate the diversity of e2e paths when only one end-system is multihomed. In addition, the reliability-oriented approaches in the literature have the following shortcomings: (i) they do not provide an online and dynamic MP selection, (ii) they do not support low unavailability levels in the order of 0.0001% based on conducted evaluations, and (iii) they do not monitor and predict the unavailability of multiple e2e paths based on only the information collected by the end-systems to ensure the availability of the communication service in the short and long terms. Lastly, direct utilization of existing MP communication protocols to achieve high communication reliability is not feasible as explained in Section 3.4.

From the above mentioned gaps in the current state of art, I propose RC4CPS. It is an e2e approach that provides an online and dynamic MP selection that chooses the minimum number of e2e paths with the highest diversity and lowest unavailability to fulfill the application desired limit on unavailability. By selecting the minimum number of e2e paths, RC4CPS targets limiting the redundant data by the required unavailability. As a result, RC4CPS reduces utilization of network resources, provides higher scalability, reduces overhead at the receiver, and might also reduce cost if the service providers charge per data volume carried.

4 Technological Background

This chapter provides an overview about a number of topics related to this dissertation. These include data communication networks and the impact of their deficiencies, middleboxes and their interactions, communication networks used in today's CPSs, MP communication and its benefits, and a short overview about smart grid applications.

4.1 Data Communication Networks

In general, a communication network is a system that allows two or more end-systems (also called hosts) to be connected and to exchange data. The term end-system does not necessarily refer to computers only, rather to any kind of equipment that is able to connect to the network. The network itself, as shown in Figure 4.1, consists of nodes and links to connect them. The end-systems connect to the network by connecting to some of the network nodes. If it is assumed that all links in the network are bidirectional, then, each node is capable of receiving or forwarding data over any of the connected links. A node can receive data from an end-system or from another node. Similarly, a node can forward data to an end-system or another node. The physical medium used to realize the links might be wired (e.g. copper wire or optical fiber) or wireless (e.g. microwave radio transmission) and might differ from one link to another. In addition, each of the links might have different capacity that is measured by the maximum bit rate provided.

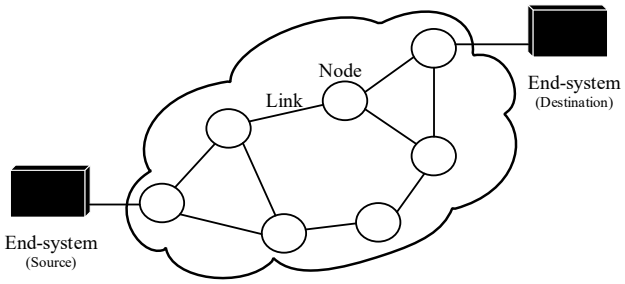


Figure 4.1 General topology of a communication network [47].

In this chapter, only data communication networks, in which the information to be carried consists of 0 and 1 data streams, are discussed. Analog communication networks, such as analog telephone networks, where analog signals are transferred without digital encoding are not considered. This is basically due to the digital nature of CPSs.

In data networks, data are carried over the network in small units, called packets, which have certain formats determined by the network. The network also specifies the maximum size of the packets and the extra information (beside the actual data) needed

to transfer them over it. The extra information includes for example the source and destination addresses and the number of bytes. Packets usually consists of a header where the extra information is included and a payload where the actual information is included. Based on the network, a packet might also include a trailer to carry part of the extra information.

The widely adopted classification of data networks is based on the area covered and the number of users served by the network. According to this classification, there are local area networks (LANs) and wide area networks (WANs). LANs refer to networks that are confined in space such as those in a single building or in a campus. In contrast, WANs refer to networks that span large geographical areas and connect two or more LANs. An example of a WAN is the Internet which is considered as the largest WAN.

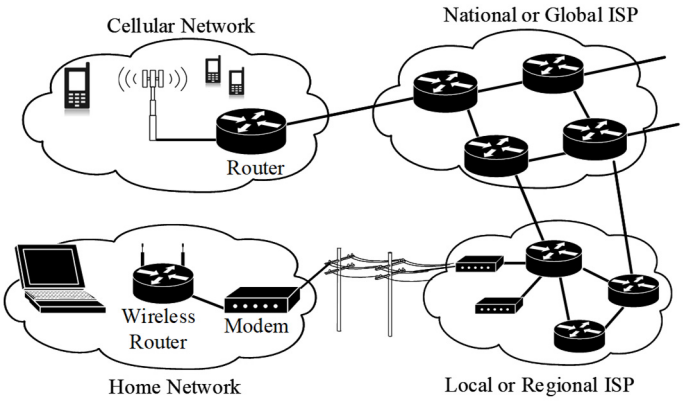


Figure 4.2 Simplified topology of a very small part of the Internet [79].

The Internet is a global data network of interconnected communication networks. It was defined by the U.S. Federal Networking Council resolution on October 24, 1995, to be a global information system that is logically linked together by a globally unique address space based on the IP protocol. In the Internet, networks and routers that are under the control of a single administrative entity are usually referred to as an Autonomous System (AS). In some cases, two or more ASs might belong to the same administrative entity. Each AS is assigned a unique number, known as the AS number (ASN), and specific blocks of IP addresses. This allows the different ASs on the Internet to acquire a way to reach each other. Hence, the Internet can be considered as a system of interconnected ASs. As shown in Figure 4.2, the architecture of the Internet can be considered to mainly consist of three levels or tiers. First, there are the access networks (access ISPs) to connect end users to the Internet using a variety of wired and wireless technologies. Second, there are the regional ISPs that connect the different access ISPs on a regional level. Third, there are the tier 1 ISPs that connect to other tier 1 ISPs and, consequently, connect subscribed regional ISPs at different regions of the world. Each of these smaller networks might have different network service provider (NSP) policies and different link and physical layer technologies, but all are connected logically using the IP protocol.

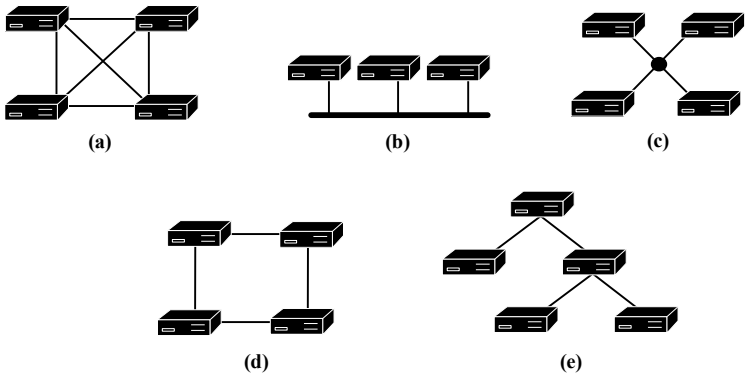


Figure 4.3 Fundamental topologies of communication networks: (a) full-mesh, (b) bus, (c) star, (d) ring, and (e) tree.

Another method to classify data networks is based on the physical topology of their components. Among the possible organizations of the physical topology are full-mesh, star, bus, ring, and tree. These topologies are illustrated in Figure 4.3. In a full-mesh topology, every host connects to every other host in the network. This topology provides high redundancy and performance, but it can be used only when the number of hosts is small. In a bus topology, a shared medium such as a cable is used to connect all hosts. Data sent by any host on the bus are received by all other hosts. Physical damages to the bus usually divide the network and isolate its different parts. As a result, bus topology is usually difficult to maintain. A widely adopted physical organization of networks is the star topology where each host in the network uses a separate link to connect to a central entity. The topology provides higher flexibility with regard to adding or removing hosts, however, a failure of the central entity results in a failure of the entire network. In a ring topology, hosts are connected in a circular fashion where each host has two neighbors. In this organization, the data travel in one direction (clockwise or counter clockwise) around the ring. Each host on the ring acts as a repeater and forwards the data to the next hop till it reaches the designated destination. A failure of a host or a link between two hosts will result in a failure of the network. Lastly, the tree topology divides the network into levels. The hosts at the lowest level of the tree, known as the leaves, can act as senders or receivers while hosts at higher levels act also as repeaters. This topology is usually used to provide cost-effective organization to connect large number of hosts (leaves). The above mentioned topologies are fundamental topologies where real networks usually combine them. For example, an ISP might adopt the ring topology for the core part of the network and the tree topology for last mile connectivity.

All kinds of communication including human face-to-face communications and network communications need predetermined rules in order to be successful. In data networks, protocols organize the different tasks between two communicating devices. For example, they define the format and maximum size of data packets, the way to begin and end communication between two hosts, and the way packets are routed between

hosts through the data network. At the beginning of networking industry, manufacturers provided proprietary equipment and protocols for networking. As the cooperation between the different companies started to increase, the need for sharing data and networks increased too. As a result, standards for networking became necessary to provide interoperability between vendors [80]. In particular, the OSI reference model and the TCP/IP model were created. The protocol stacks of the OSI and TCP/IP models are illustrated in Figure 4.4.

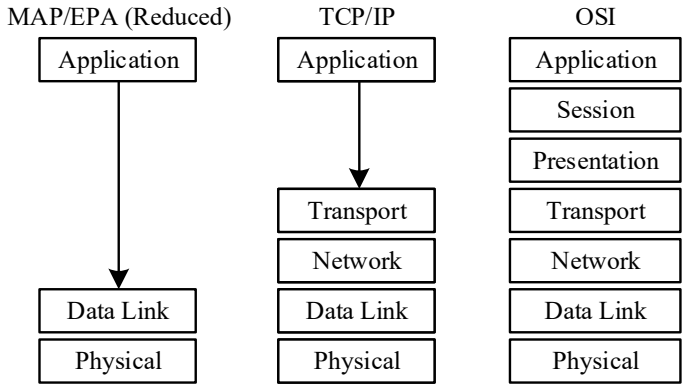


Figure 4.4 Protocol stacks of OSI reference model, TCP/IP, and reduced MAP/EPA model.

As illustrated in the figure, both models are layered models to tackle the complexity of describing network communications. This is also the adopted approach in implementing the communication process in real networks. Each layer in these models provides one or more services to the layer above it. Therefore, the different protocols that perform specific functions or tasks in the entire communication process are grouped into the different layers of these reference models. In contrast to the OSI model which describes the communication process in general, the TCP/IP model considers only the TCP/IP protocol suite and its communication process. A brief description for each layer of the OSI model is given as follows:

Application layer: Provides services for user applications to access the network.

Presentation layer: Provides information to the application layer with regard to the data format.

Session layer: Establishes, manages, and terminates sessions between users applications.

Transport layer: Performs data segmentation and numbering at the source, data transfer, and data reassembly at the destination.

Network layer: Creates data packets and addresses them for e2e delivery in a multi-node network.

Data link layer: Creates data frames and address them for delivery between nodes that share a physical layer.

Physical layer: Transmits and receives binary data symbols over a physical media.

4.2 Impact of Communication Network Deficiencies

Data networks exhibit by necessity performance limitations and reliability limitations. Performance limitations are mainly caused by the nature of the physical media used. By contrast, reliability limitations are attributed to several reasons including components and network failures (caused by, for example, procedural errors and software or hardware updates), oversubscription, environmental conditions (e.g. effect of weather on wireless communications), slow recovery of routing protocols when communication paths fail, and power disruptions. As a result, some deficiencies with regard to communications performance and reliability arise. The major performance deficiencies are described first and are as follows:

Time delay: Which is the average time required for the delivery of data packets between the source and destination end-systems in the network. This time delay depends on several factors including (1) the media access scheme which determines the time taken to accept a new packet by the network; (2) the transmission time of packets inside the network. The later factor depends in turn on the propagation time of signals over the network medium and the queuing and processing times of network components.

Jitter: If the time delay introduced by the communication network is variable, the arrival time of packets will fluctuate from one packet to another, which is known as jitter. This is mainly attributed to the highly varying nature of network conditions such as network load (e.g. congestion) and the quality of communication channels or links.

Packet loss: Another communications deficiency is the loss of data packets due to several reasons including transmission errors in error prone channels (e.g. wireless channels) and buffers overflow of network devices during congestion.

Limited bandwidth: Communication channels of data networks have finite capacity, which is mainly caused by the limited capacity of physical media used. As a result, the data transfer rate over communication networks is also limited. Also, data transfer rate is limited because data networks are shared between different components and applications.

The above-mentioned performance deficiencies were considered in [81] to be sufficient metrics to describe the provided quality of service (QoS) by a computer network (network performance). Where QoS, as defined in [82], [83], is the user satisfaction determined by the collective impact of service performance. Such QoS is practically represented by a set of performance metrics such as average time delay and provided data rate to give a mean to specify required performance. It was also indicated in [84] that improving QoS, in general, requires minimizing the time delay. While the effect of the other metrics such as jitter can be reduced utilizing existing approaches that provide a tradeoff between these metrics and time delay. Therefore, and due to the limited scope, only the impacts of time delay are considered in more details.

As CPSs incorporate different control systems, the time delay in the control loops of these systems has a significant impact on control performance. Here, system stability is a key control performance parameter and depends on the response time of the system defined as the maximum time allowed between the occurrence of an event and applying the corresponding reaction [85]. When a control loop is closed using a communication network, then this will entail performance degradation or even destabilization of the system [86] due to the presence of time delay and other communication deficiencies. In this context, the traffic of CPSs, even between their units, is usually characterized as a RT traffic [4]. The notion of RT traffic means that the traffic has some sort of an upper bound or deadline on information delivery delay. This notion can be further featured, depending on the effect of the deadline violation on the control system, to be either soft or hard [85]. If missing the deadline will mark the late information as useless or even negatively impact correct system operation, then the deadline is hard. If otherwise the late information will degrade performance efficiency of the system without jeopardizing its operation correctness, then the deadline is soft. Consequently, RT systems are usually classified to be soft RT or hard RT systems.

The importance of time delay for systems performance can also be observed from the required network performance for CPSs. Over the last decade, many CPSs applications have been proposed along with studies estimating their traffic characteristics and/or communication requirements [13], [87], [88]. In these studies, time delay was considered as the key performance metric of communication networks in order to realize the proposed CPSs. Moreover, it was clear that all proposed CPS applications require an upper bound on the communication delay rather than a fixed value. Consequently, the effect of jitter on such systems was not considered as long as the information delivery deadline is not violated. Indications on the needed data rates for such CPS applications were also provided. Other network performance metrics such as maximum allowed packet error and loss rates were not considered in many of these studies. However, the presence of such communications deficiencies will certainly degrade CPSs performance [89].

Similarly, reliability deficiencies of communication networks can also negatively impact control loops stability or even stop their operations. For example, when a frequency event in a power grid occurs, such as a sudden loss of generation, the grid frequency response is divided into three phases [90]. One of these phases is the automatic generation control (AGC). In this phase, the grid utility operator sends power signals to the different power plants to adjust the level of generated power and restore the grid frequency to its nominal value. The time frame for the AGC to occur is between five and ten minutes. If it is assumed that the communication service between the utility control center and the generation planets is unreliable, which might happen due to failures of routing protocols. In that case, the time frame to apply the AGC cannot be met with high probability. This, in turn, might prolong the duration of the frequency

event and cause damage to customer appliances that are designed to work at a certain grid frequency. Another important issue to indicate here is the requirement on network performance. In this example, it is clear that the performance requirements on the communication service are low and a time delay of several seconds, as an example, can be tolerated. On the other hand, communication service reliability is more critical for such application.

Another important issue regarding communication reliability is the data transfer reliability which refers to the delivery of messages to the intended recipient(s) complete, uncorrupted, and in the order they were sent [81]. In public data networks such as the Internet, data transfer reliability is mainly deteriorated by congestions. In such networks, achieving reliable data transfer is almost left to the end-systems, for example, by utilizing TCP transport layer protocol. Network-based approaches to improve the data transfer reliability start to appear in newer communication technologies such as the Universal Mobile Telecommunications System (UMTS) [91] and Software-Defined Networking (SDN) [92]. These approaches are mainly based on differentiating and classifying users' traffic and associating it with certain forwarding treatments (e.g. resource allocation, prioritization, and packet error loss rate). However, similar approaches are not feasible in the Internet due to the high heterogeneity of connected ASs and their corresponding networks. More specifically, each network has different reliability and performance limitations. In addition, each network might have different policies for traffic forwarding and utilize different technologies. To use such solutions in the Internet, cooperation across different ASs, networks, and components to provide the same forwarding treatments is required.

4.3 Middleboxes

In the 70's, during the early years of the Internet, the United States Defense Advanced Research Projects Agency (DARPA) developed the TCP/IP suite as part of the ARPANET to meet the needs of an open-architecture network environment. ARPANET was one of the first designs of nowadays commonly known networks [93]. Throughout the years, the Internet protocol suite evolved into a family of important networking protocols which slowly became standardized with the expansion of the commercial Internet and the growth of private networks. Among others, it contains the TCP and the UDP protocols, both responsible for the delivery of data in networks using the IP protocol. While TCP provides an ordered, reliable, and error-checked data stream between separate host applications [94], UDP provides a lightweight, connectionless datagram service with a focus on reduced latency rather than reliability. Together with ICMP, used for diagnostic or control purposes, they form the backbone of today's Internet. They are also expected to be fully supported in their standard configuration in every network infrastructure based on TCP/IP stack.

Initially the TCP/IP architecture was designed to follow the e2e principle, which proposes a passive network that should not interfere with mechanisms provided at the application layer [94]. It was assumed that packets would flow from source to destination unchanged. The Internet, however, evolved differently and today it consists of countless interconnected public, private, academic, business and government networks. These networks include millions of intermediary devices whose functions differ from standard IP routers, switches and repeaters. They introduce dependencies and hidden points of failure by manipulating the contents of IP packets outside of the application layer. As a result, they violate the basic idea of the e2e principle [95]. These devices are referred to as middleboxes [14]. With this regard, a middlebox is defined as any intermediary device performing functions other than the normal standard functions of an IP router on the datagram path between a source host and destination host. Middleboxes come in many forms and with a wide range of different functionalities besides usual IP forwarding. While TCP and IP headers provide space for additional options and generally have a lot of potential for extensions [96], [97], the deployability of these can be very difficult due to the impact that middleboxes started to have on traffic between remote networks.

In the following two sub-sections, only firewalls and network address translators (NATs) will be described, as they are the most commonly used middleboxes in home and enterprise networks. Proxies, load balancers and intrusion detection systems are other mentionable common middleboxes [14].

4.3.1 Firewalls

Business networks all over the world rely on firewalls to control out and ingoing traffic, and delimit their networking environment from the Internet. Even home routers mostly have these and other middlebox functionalities integrated [98]. In addition, modern operating systems (OSs) like Windows deploy application firewalls and enable them by default. These and many other firewalls often work based on a white list approach, through which only explicitly allowed communication flows are able to pass. Anything that is not defined on that list is forbidden and, as a result, the corresponding packets are discarded. This leads to a hurdle for any kind of traffic that attempts to use protocols besides classical TCP, UDP and ICMP. But even regular TCP/IP can run into problems when facing stateful firewalls. Stateless firewalls used to treat packets or network frames individually. By contrast, stateful firewalls consider also the packets context by validating sequence numbers, ports, and IP addresses and match them with any known and active connections [99]. The intention is to filter illegitimate packets like they occur in malicious injection attempts. This is an important detail that needs to be considered when working with MP protocols. This is because their design often makes use of additional data flows with non-continuous sequence numbers. An example of how nowadays firewalls increase the complexity of modifying existing protocols was presented in [71]. The example considered modifying the receive window in TCP

header for better performance in high bandwidth networks. This TCP option is considered standardized and was defined in 1992 [96], nevertheless, today's firewalls impose a major limitation on exploiting TCP specification. Medina et al. [95] further illustrates how the use of various TCP and IP extensions can lead to major QoS degradation. More specifically, the deployment of additions such as Explicit Congestion Notification (ECN) [100] (enables routers to mark packets and notify the sender about congestion) and Path Maximum Transmission Unit Discovery (PMTUD) [101] (allowing TCP to determine the largest possible segment size for the current communication path using ICMP) showed that only a minority of the tested web servers actually responded as intended. The reasons for this were: (i) middleboxes cleared or malformed the additional marks in the headers, (ii) middleboxes refused or reseted the connection, or (iii) middleboxes, in the case of PMTUD, entirely blocked ICMP packets on which the mechanism relies on. Any Internet path that removes unknown options will not allow the deployment of TCP extensions.

Similar results were discovered with IP options where all of the tested TCP connections failed or ignored the options that were completely unknown. This would also be the case of any new design. This goes back to the fact that in most routers, the basic task of IP forwarding is done by hardware for efficiency reasons. Packets that carry IP options are considered as an exception and are processed by software. However, to protect routers from denial-of-service attacks, these packets get dropped in most cases [71].

4.3.2 Network address translators

NATs are another widely deployed middlebox. Their purpose is to let multiple hosts in a group share a single IP address. This way a network of users can use private addresses internally and let a NAT map them to a single public address when accessing remote networks like the Internet [102]. The motivation behind this technology is to preserve IPv4 addresses, hide network topologies, obscure host addresses and become less dependent from ISPs. Besides commercial networks, home users also deploy NATs through their home routers. They receive a single public IP address from their ISP and a NAT inside their routers maps it to all wired and wireless devices in the house. A study in [103] showed that only 10-20% of the tested peers were directly connected to the Internet without a NAT. The functionality of NATs is based on transparently manipulating TCP/IP headers for in- and outgoing segments. To correctly forward them, the source/destination addresses and ports need to be rewritten. Since these entries are included in the TCP and IP header checksums, they need to be updated too. Due to the changes done to the segments payloads, these middleboxes are referred to as content-modifying middleboxes. If a protocol is not supported or uses unknown semantics, as it can be the case with new extensions, its packets cannot be translated correctly and are lost or discarded. Therefore, new or modified protocols need to consider the way NATs work, in order to work properly in the presence of such middleboxes.

4.4 Communication Networks for CPSs

LANs and WANs can be categorized, based on the domain, to industrial and general purpose networks. The majority of communications in today's CPSs are realized using industrial communication networks to fulfill the RT communication requirements. Local industrial control networks, as illustrated in Figure 1.1, are used extensively within the control subsystems of a CPS. The motivation for such networks is to replace the point-to-point communication at the plant level between the different field devices (e.g. sensors and actuators) and their corresponding controllers (e.g. programmable logic controllers (PLCs)) by multipoint-to-multipoint communication (bus). Local industrial networks are usually confined in space and differ from general purpose networks as the requirements of such networks are much higher. As a result, they are capable of providing RT communication, predictable throughput, and very low down times, and can operate in harsh environments (e.g. high noise environments). In addition, the data packets over such networks are characterized by having small sizes with low protocol overhead and the topology of such networks can take different forms (e.g. star, ring, tree, etc.) depending on the application. Compared to the OSI network model with 7 layers, most of the local industrial networks are based on the Manufacturing Automation Protocol/Enhanced Performance Architecture (MAP/EPA) reduced network model. As shown in Figure 4.4, the MAP/EPA model consists only of the application, data-link, and physical layers [104].

A wide variety of local industrial networks were developed over the last few decades, called fieldbus systems. These systems, with the majority standardized in the IEC 61158 [105] standard, were proposed for different industrial markets and offer different features. Examples include Controller Area Network (CAN), PROcessFIeld Bus (PROFIBUS), INTERBUS, and Factory Instrumentation Protocol (FIP). In recent years, newer Ethernet-based fieldbus standards were proposed such as Ethernet for Control Automation Technology (EtherCAT), Process Field Net (PROFINET), or Ethernet/IP. This is mainly due to the technological advances in Ethernet which allowed new features including RT communication capabilities, high data rates, full-duplex data transmission, and low congestion with the use of switched networks.

Existing industrial WAN networks today are mainly used by the Supervisory Control and Data Acquisition (SCADA) systems to monitor and control remote industrial infrastructures. Such systems usually consist of remote field devices such as remote terminal units (RTU) or PLCs that connect the remote components (e.g. sensors) to the WAN network of the enterprise (e.g. utility operator). The field devices at the remote sites connect in their turn to a central supervisory computer with a human-machine interface. The intermediate layer of field devices between sensors and communication infrastructure allows the digital transmission of sensor signals using industrial communication protocols. Almost all SCADA WANs are using private networks (fiber-optic or radio links built specially between the sites of the system) or dedicated leased

lines from national WAN operators to connect the different remote sites of the enterprise. Figure 4.5 shows a simplified view of SCADA system for remote site monitoring. For simplicity, additional components such as modems were not included in the figure.

One example of the common SCADA protocols used between the field devices and the master controller represented by the central supervisory computer is the Distributed Network Protocol (DNP3) [106] standardized in the IEEE 1815 standard [107]. DNP3 allows the master controller to poll field devices for measurement and status information either periodically or on demand and also to send commands to these devices. Similar to industrial LANs protocols, only the application, transport, and data link layers of the OSI model are present in DNP3 with a maximum transport protocol data unit (PDU) of 250 bytes. The protocol is defined on serial connections at the physical layer (e.g. using RS232 standard) and supports IP-based networks by adding a fourth layer below the data link layer. This layer is called the Data Connection Management and allows the DNP3 with its layers to be the application layer of popular transport protocols such as TCP and UDP.

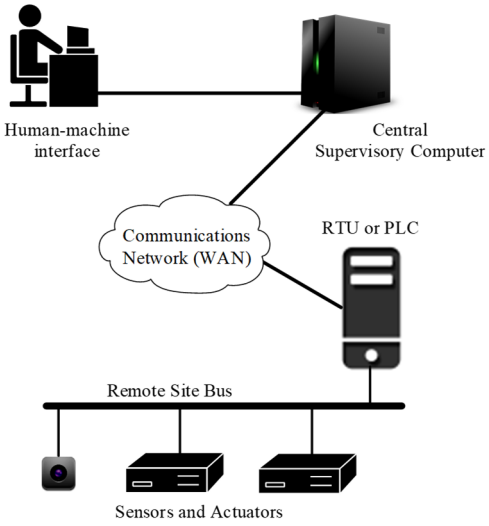


Figure 4.5 Simplified diagram for remote site monitoring using SCADA.

Compared to local industrial networks, SCADA systems are usually considered as slow response systems where the response might take several seconds and/or requires the intervention of a human operator [47]. The response time of SCADA systems is higher due to several facts including: The nature of existing SCADA protocols such as DNP3, the need for special hardware to allow their transmission over prominent WAN networks (e.g. IP-based networks), and the use of limited bandwidth links to reduce networking expenses. As a result, SCADA systems are usually considered to perform coordination rather than direct RT process control.

Due to the unpredictable timing behavior of general purpose LANs, they are not used at the field level of control between PLCs, sensors, and actuators. However, such networks

might be used at the supervisory level to monitor and coordinate the different processes within a plant. In this level, the RT and determinism requirements are much less stringent. Similarly, general purpose WANs can also be used for CPS applications with low RT and reliability requirements, but with the components distributed over a large geographical area [47]. Moreover, new wireless standards such as the Long-Term Evolution (LTE) provide very low latencies (as low as 50 ms), high data rates, and very comprehensive framework for QoS. This increased the interest in utilizing them for CPSs instead of using private WANs. However, the existing usages of such technology have several drawbacks that limit their benefits to CPSs. It is also worth mentioning here that upcoming wireless communication standards, denoted by the 5th generation (5G) [108], consider the communication requirements of CPSs. More particularly, the requirements of 5G networks as defined by the Next Generation Mobile Networks (NGMN) Alliance include support for enormous number of concurrent connections (e.g. to support large deployments of sensors), very low e2e latency in the order of 1 ms, enhanced coverage, and improved spectral and signaling efficiencies compared to the 4th generation technologies (e.g. LTE).

CPSs were envisioned to use the Internet as their future communication network. This is also motivated by the technological advances in communication technologies that enabled high-speed data communications. However, Internet utilization for CPSs is limited by factors such as the high heterogeneity of commercial networks constituting it, the different operator and QoS policies, and the insufficient communication reliability provided by it (More details are provided in Chapter 3).

4.5 MP Communication

MP communication in this dissertation refers to the use of more than one e2e path between end-systems for communication. To ensure resilience, path diversity was taken into account in the designs of the core parts of the Internet. Only routers in the early years had multiple network interfaces. Hosts used only one physical interface and transport protocols were built for single-path e2e transmissions. Traditional TCP/IP does not natively offer MP support; therefore, most of today's communication is based on single-path communication. However, the number of multihomed devices has significantly increased over the last decade. Smart phones are equipped with WiFi and 3G or 4G interfaces, Laptops have WiFi and Ethernet interfaces, and servers, especially those in data centers, are accessed via multiple interfaces. Moreover, the hardware and software platforms of hosts and servers are designed to allow the addition of extra network interfaces. So even though that the resources are there, they are rarely utilized by the applications. For instance, mobile devices can switch from one communication service to another based on its availability. These services are not used concurrently rather in a failover-fashion and, consequently, the e2e communication stays single-path. As a result, the interest for MP communication in the Internet is increasing. This is

attributed to the several benefits of MP communication. Below, a brief description of the major benefits obtained using MP communication is provided [109]–[112]:

- Reliability and fault tolerance – The utilization of multiple paths within a communication session can improve the fault tolerance and reliability. Redundant information can be routed to the destination through multiple instances. In case of a link or node failure on one path, redundant data from other paths can be used. Unlike the case of single-path communication where higher delays are caused by failure-triggered route discovery, MP communication enables the use of active backup routes.
- Load balancing – Over-utilized links can cause congestion and therefore induce packet drops and delays. If multiple paths exist, less congested paths can be used to divert the traffic and ensure a balanced overall load throughout the network. The idea follows the Resource Pooling (RP) principle described in [113] which suggests to consider a collection of networked resources to be one pooled resource.
- Bandwidth aggregation – when there are several low-bandwidth links available for a node and better performance is needed, then, the traffic can be splited into multiple flows over multiple paths. This way an improved throughput is obtained which potentially equals the throughput of one path times the number of used paths.

4.5.1 Path Diversity and Disjointedness

MP communication benefits are the highest when the used paths are disjoint. This is because disjoint paths are characterized by uncorrelated metrics such as time delays. By pooling such diverse paths and their corresponding resources, reliability can be significantly increased in the Internet. For applications, MP communication provides a predictable average behavior compared to single-path transmissions. In the case of disjoint paths, it is unlikely that all the paths will experience congestion concurrently. With backup paths to set-off disturbances, the communication session reliability can be maintained relatively constant.

The Internet is a very large network and represents the most diverse network on the globe. Hence, it is expected to be the most promising network to utilize MP communication.

4.6 Smart Grid Applications

Smart grid was defined in [114] as a distributed and automated network for energy delivery that provides bidirectional flow of data and electricity and allows achieving near-instantaneous balance between supply and demand by combining the advantages of distributed computing and communications. As described in [47], the vision of smart

grids targets modernizing existing grids by adopting recent technologies from the different domains to improve reliability, security, and efficiency. Another objective of smart grids is the reduction of harmful emissions in the environment. This requires incorporating renewable energy sources in the grid, efficient energy management, and active participation of individuals in energy management. Different applications were proposed to achieve the goals of smart grids. Examples include advanced metering infrastructure (AMI), demand response (DR), distribution automation (DA), and wide-area voltage stability monitoring (WAVSM). AMI refers to an infrastructure of smart meters and communication technologies to exchange usage information between individuals and utility operators. The collected information is used by the utility operator for different purposes such as billing to the individuals and grid management. DR refers to achieving the balance between demand and supply by either increasing the power supplied to the grid or by reducing the demand on the grid. The methods for DR include dynamic pricing and the voluntary participation of customers by allowing the utility operators to directly control the load of appliances and thermostats. DA refers to the remote monitoring and control of the assets in the distribution part of the grid by relying on automated decision-making. Therefore, DA requires increasing the intelligence of the distribution side of the grid through the use of, for example, intelligent electronic devices (IEDs). These devices are microprocessor-based controllers in power systems that are used to control the different equipment such as circuit breakers. WAVSM refers to the use of a set of technologies to monitor the power grid health in real-time and enable fast response initiation when abnormalities are detected. High speed response would protect the power grid and prevent service disruptions events such as black-out events over wide service areas.

Detailed description of the above-mentioned applications is provided in the literature presented in Section 3.1.1 (e.g. in [47]). In the next section, an application scenario in smart grids is described and the related communication requirements are considered.

4.6.1 Exemplary Scenario for a Smart Grid Application

With the introduction of renewable energy resources such as wind turbines, future smart grids are expected to heavily rely on wide-area monitoring applications. The essential components for such applications are the Phasor Measurement Units (PMUs) which conduct precise measurements regarding the power grid state represented by the current and voltage phasors at the corresponding locations of the units. In this section, only the communication requirements to deliver the data packets generated by the PMUs to the utility operator are considered. Other details such as the manner in which the phasor readings are calculated and used are out of the scope of this work. The PMU data frame structure along with other communication specifications are provided in the IEEE C37.118 standard [24]. With the assumption of having one digital field, two analog fields, four phasor readings and UDP/IP protocol stack as in [87], the overall MAC protocol data unit size is 76 byte. The standard also specifies the PMU reporting

frequency for 60 Hz power systems as 10, 12, 15, 20, 30, and 60 Hz and for 50 Hz systems as 10, 25, and 50 Hz. Support for other reporting rates is also encouraged by the standard. The North American Synchro-Phasor Initiative (NASPI) defined five classes of phasor data services, namely classes A through E [65]. Examples for applications of these data service classes include wide-area voltage stability monitoring (WAVSM), data visualization, and post event analysis. Each class has specific requirements on the communication service availability and performance. The availability requirements range between 99% and 99.9999% while the maximum latency requirements range between 50 ms and 2 min.

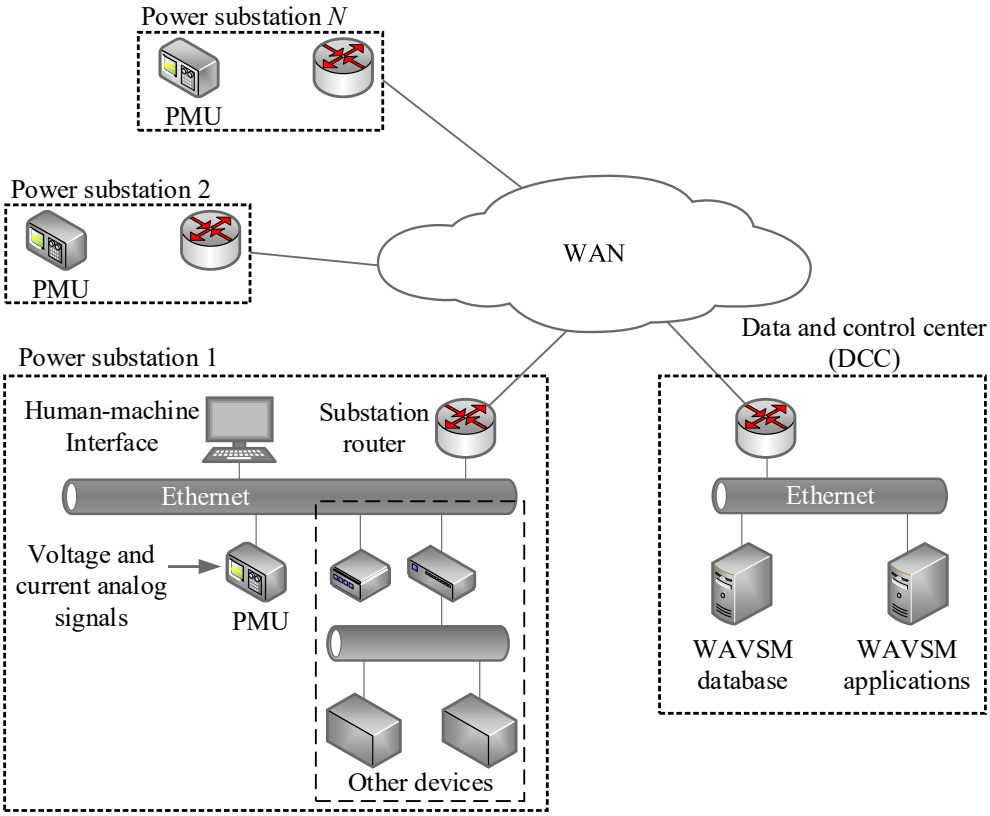


Figure 4.6 Simplified scenario for a smart grid application with PMUs sending data over a WAN to the utility operator.

In Figure 4.6, a simplified scenario for the WAVSM smart grid application is illustrated. The scenario considers transmission substations and a data and control center (DCC). In power grids, transmission substations connect transmission lines and perform different functions such as conversion of transmission voltages between transmission lines. The transmission lines carry the electrical energy from generating plants to the distribution substations where the customers are connected to the grid. The PMU data is transmitted from each transmission substation to the DCC over a WAN connection. The WAN is expected to satisfy the communication requirements of the WAVSM application and its corresponding data service class. Within the transmission substation, the PMU connects using Ethernet to a router that connects the substation to the WAN. The PMU data is

used by the WAVSM applications at the DCC to determine the current power margin with reference to voltage stability [116]–[118]. A power margin is the quantity of additional power that can be transferred without jeopardizing voltage stability of the transmission system. Such information allows utility operator to determine future actions such as generation rescheduling to maintain voltage stability. At the DCC, a database might be utilized to store the PMU data for research and development purposes.

As indicated in [37], typical message size for measurements made by a PMU for WAVSM is larger than 52 bytes. Also, the typical data sampling is once every 0.5–5 s and the delay for delivering the data samples should be less than 5 s. Lastly, the WAVSM application requires an availability of communication service higher than 99.9%.

5 Reliable Multipath Communication for Internet-based CPSs (RC4CPS) – Concept

In this chapter, the concept of the proposed approach RC4CPS is detailed. As mentioned previously, RC4CPS is an approach to provide reliable communication for Internet-based CPSs. The approach considers the desired reliability level imposed by the application which is translated to the percentage of time that the network is required to be available. RC4CPS provides a dynamic online MP selection to fulfill the required availability. It also provides online monitoring to determine the attributes of the different paths and their combinations. The MP selection considers also e2e paths diversity and future unavailability probability to maximize the gains of MP communication. In the following sections, the considered system model, the optimization formulation and selection metrics, and the architecture of RC4CPS are introduced.

5.1 System Model

In the considered CPS model: (1) the units of the CPS that need to communicate over the Internet are geographically separated; (2) each communicating component uses two or more network interfaces where each network interface is connected to a different access network (access ISP). A simplified diagram of two components is shown in Figure 5.1 in which one e2e path exists between each pair of source-destination network interfaces. If the set of source and destination interfaces are denoted by $S = \{1, \ldots, n\}$ and $D = \{1, \ldots, k\}$ respectively, then each path is represented as a pair (i,j), in which case $i \in S$ and $j \in D$.

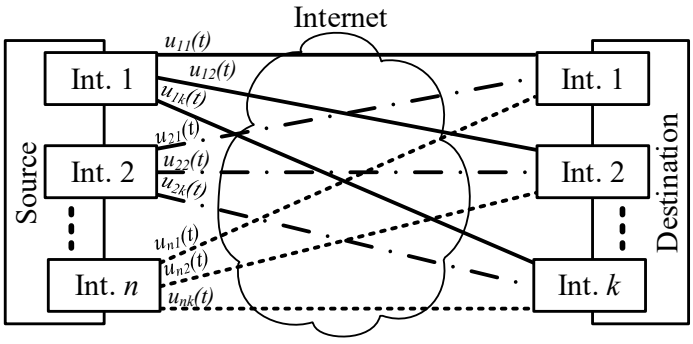


Figure 5.1 System model for MP communication between two CPS components using different e2e paths.

In order to measure the unavailability of a certain e2e path (i,j) over a period of time T, test packets are sent every t seconds like in [11], [48]. Here, (i,j) is available when the source can use it to successfully send a test packet. A successful test packet transmission is indicated by receiving its corresponding reply within the specified timeout interval. If

the packet is lost or its corresponding reply was received after the specified timeout, (i,j) is considered unavailable. Hence, and based on the results of the probes over period T, the approximated instantaneous unavailability function of (i,j), denoted by $u_{ij}(t)$, is given by:

$$u_{ij}(t) = \begin{cases} 1 \text{ when } p \text{ is unavailable at time } t \\ 0 \text{ otherwise} \end{cases} \quad (5.1)$$

The approximated instantaneous availability function of (i,j) can be obtained using (5.1) as $a_{ij}(t) = 1 - u_{ij}(t)$. Hereinafter, the approximated instantaneous unavailability function is referred to simply as the instantaneous unavailability. Communication service unavailability is usually measured as the proportion of time that the service is unavailable for use to the total time [119]. Such measure is referred to sometimes as the (average) interval or mission availability [9] or as the average uptime availability [120] which is due to the different classification adopted in the literature [121]. This measure will be referred to hereinafter as the average unavailability or simply as the unavailability. For a path (i,j), the average unavailability is given as:

$$u_{ij} = \frac{1}{T}\int_0^T u_{ij}(t)\, dt, \quad (5.2)$$

where $u_{ij} \in [0,1]$ (which is not time dependent as in the case of $u_{ij}(t)$). For MP communication, the unavailability of a set of paths $\theta = \{(i,j) \mid i \in \{1, \ldots, n\}, j \in \{1, \ldots, k\}\}$, where each path is represented by the pair of interfaces using it, is needed. In the case that the different paths in θ can be treated as independent, the unavailability of the multiple paths is stochastically expected to be:

$$u(\theta) = \prod_{(i,j)\in\theta} u_{ij}, \quad (5.3)$$

in which case $u(\theta) \in [0,1]$. This point out the potential of MP communication, but can be applied only if the unavailabilities of the paths do not influence each other. Equation (5.3) is not used because RC4CPS is expected to utilize subsets of available e2e paths where some of the paths might be dependent. In addition, the diversity estimation mechanism adopted by RC4CPS (Section 5.2.1) does not provide a deterministic answer about dependency (dependent or independent). Since independence cannot be assured at this step, the time dependent unavailabilities given (5.1) are used to determine $u(\theta)$ in RC4CPS and is given as:

$$u(\theta) = \frac{1}{T}\int_0^T \prod_{(i,j)\in\theta} (u_{ij}(t))\, dt. \quad (5.4)$$

This represents the proportion of T where all paths in θ are simultaneously unavailable. For instance, if $\theta = \{(1,1), (2,2)\}$ with instantaneous unavailability functions of its paths as shown in Figure 5.2a. Then, calculating $u(\theta)$ over a period T as given in (5.4) requires first multiplying the instantaneous unavailabilities of the two path as illustrated in Figure 5.2b and dividing the integral of the product by T.

For the communication service to fulfill the reliability requirements of the CPS, the unavailability of the communication network or simply the communication path(s) needs to be less than the maximum allowable unavailability u_r, where $u_r \in [0,1]$.

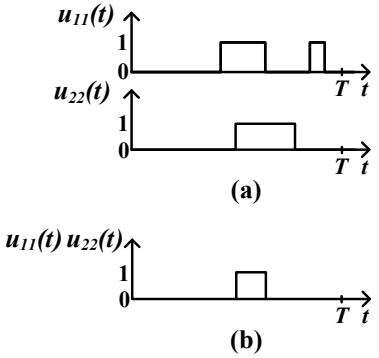


Figure 5.2 MP unavailability ($u(\theta)$) calculation of two paths: (a) Instantaneous unavailability and (b) MP instantaneous unavailability.

5.2 Optimization Formulation for MP Selection

According to sub-objective 2 of this research, the approach proposed targets minimizing the number of e2e paths required to provide a certain level of communication reliability. Therefore, the MP selection problem is formulated as an optimization problem that consists of selecting the minimum set of e2e paths with the highest diversity and lowest future unavailability probability. This is motivated by the results from the conducted measurements in Chapter 6 to answer the first research question. It was observed that unavailability events usually follow certain patterns and paths traversing different networks have lower unavailability. For completeness of view, the approaches for diversity estimation and unavailability prediction are first introduced. Then, the optimization formulation for MP selection is described.

5.2.1 MP Diversity

It is necessary to estimate the diversity using e2e measurements only as RC4CPS will run at the end-systems. For RC4CPS, it is not required to select absolutely independent paths but rather diverse paths as far as possible. For this purpose, I follow the procedures in [70] that use the comparison test [122] for diversity estimation. Namely, the correlation of round-trip times (RTTs) of two paths is used to estimate diversity. The use of RTTs in this dissertation is to avoid the need to synchronize communicating end-systems and the extra data processing needed to take off clocks skew. The later evaluations in this dissertation show that the use of RTTs revealed also the investigated characteristics. Besides that, e2e performance is what applications experience.

For correlation calculation, the sets d_{ab} and d_{mn} are defined as the delay values of the paths (a,b) and (m,n). The delay values are observed over the same time interval T and

nearly at the same time. After that, the sample correlation coefficient of the sample sets d_{ab} and d_{mn} is used to calculate the correlation such that:

$$\rho(d_{ab}, d_{mn}) = \frac{C(d_{ab}, d_{mn})}{\sqrt{C(d_{ab}, d_{ab})}\sqrt{C(d_{mn}, d_{mn})}}, \tag{5.5}$$

where C is the covariance of two random variables and (5.5) is the Pearson's correlation.

According to the comparison test, the correlation coefficient between the sample sets of two random variables is denoted as M_x while that of the same sample set is denoted as M_a. Using these two measures, the comparison test determines if two paths share one or more bottleneck as follows: (i) Calculate $M_x = \rho(d_{ab}, d_{mn})$ between pairs of samples obtained from d_{ab} and d_{mn} with send times separated by $T_x > 0$. (ii) Calculate $M_a = \rho(d_{ab}^1, d_{ab}^2)$ between pairs of interleaving samples obtained from d_{ab} with send times spaced by $T_a > T_x$. (iii) Two paths share a bottleneck if $M_x > M_a$, provided that packets' temporal spacing on different paths (T_x) is less than the one on each path (T_a). This conclusion is motivated by the observation that packets traversing joint paths will suffer similar impacts expressed by correlated time delays. As proposed in [70], the case with $M_x \leq M_a$ does not mean that the two paths are disjoint and they might share links. Nevertheless, the absolute value of M_x can reflect the degree of correlation and can be used to quantify the degree of diversity between paths. For a subset θ, the sum of absolute value of M_x for all possible path pairs is used in [70] to quantify MP correlation and is given as:

$$\rho(\theta) = \sum_{(i,j)\in\theta} \sum_{(m,n)\in\theta} \mathbf{1}_{(i,j)} \cdot \mathbf{1}_{(m,n)} \cdot \left|\rho(d_{ij}, d_{mn})\right|, \tag{5.6}$$

where $\mathbf{1}_{(i,j)} \in \{0,1\}$ is an indication variable that equals 0 if $i = m$ and $j = n$ and 1 otherwise and $\rho(\theta) \in [0, \binom{|\theta|}{2}]$. The form of (5.6) is mainly because *P*earson's correlation is a measure between two random variables only. Nevertheless, a set of highly correlated paths will have a higher $\rho(\theta)$ compared to another one with slightly correlated paths. Hence, the same method to quantify MP correlation will be used here.

5.2.2 MP Future Unavailability

A class of random processes that is widely used to model Internet paths is Markov Chains (MCs) (described in detail in Section 7.1). Such models are usually described by the transition probabilities between their states given as:

$$TM = \begin{bmatrix} p_{00} & \cdots & p_{k0} \\ \vdots & \ddots & \vdots \\ p_{0k} & \cdots & p_{kk} \end{bmatrix}, \tag{5.7}$$

where p_{xy} is the transition probability from state x to state y of the MC. MCs can also be represented graphically using state diagrams. For example, the state diagram of the Gilbert model, one of the simple MCs where k in (5.7) equals one, is shown in Figure 7.2. The two states in the figure might represent the occurrence and non-occurrence of

events such as unavailability events. Given the model of an e2e path, the prediction of future unavailability starts by path monitoring to determine the transition matrix (*TM*) of its model and its current state (available/unavailable). After that, the probability of unavailability occurrence during the next transmission is determined. For example, if the Gilbert model is assumed with the current state 0 (path (*i*,*j*) is available). Then, p_{01} is the probability that the path switches to state 1 (path (*i*,*j*) is unavailable) for the next transmission. Beside the average unavailability u_{ij} in equation (5.2), there is also the unavailability probability during the next transmission:

$$u_{ij}^{*}(t+\Delta t)=\begin{cases} p_{01} \text{ if the path is available at time } t \\ p_{11} \text{ if the path is unavailable at time } t \end{cases} \tag{5.8}$$

that takes the actual state of a path (*i*,*j*) into account. For multiple paths, the sum of $u_{ij}^{*}(t+\Delta t)$ of the individual paths, denoted as $u_{t+\Delta t}(\theta)$, will be used and is given as:

$$u_{t+\Delta t}(\theta) = \sum_{(i,j)\in\theta} u_{ij}^{*}\left(t+\Delta t\right), \tag{5.9}$$

where $u_{t+\Delta t}(\theta) \in [0, |\theta|]$. A better way to quantify MP unavailability probability is to use the chain rule of probability. However, the prediction process will become very complicated due to the following. First, there is a need to determine the conditional probability for each path in each subset ($|\theta|$ - 1 calculations for conditional probabilities are needed). Second, the measure will not significantly differentiate subsets (of the same size) with bad paths from those with all good paths. Equation (5.9) is less mathematically motivated. Nevertheless, it seems practically to be a suitable measure for MP selection.

At this point, it might seem that RC4CPS prediction of future unavailability is static. However, if the statistics of the monitored path will change, such changes will be present in the path history. This history is periodically used to update the path model parameters. Therefore, it is expected that RC4CPS prediction of future unavailability will reflect future changes in the path characteristics.

5.2.3 MP Selection

In Figure 5.1, the set of source interfaces is defined as $S = \{1, 2, \ldots, n\}$ and the set of destination interfaces is defined as $D = \{1, 2, \ldots, k\}$, where S and $D \geq 2$. Hence, the set of e2e paths, denoted by *P,* is the Cartesian product of *S* and *D* such that:

$$P = S \times D = \{(i, j) \mid i \in S,\ j \in D\}. \tag{5.10}$$

The Power set of *P,* denoted as $\mathcal{P}(P)$, includes all $2^{|S|\cdot|D|}$ unique subsets of *P*. In other words, $\mathcal{P}(P)$ is the set of all possible solutions, where each solution is a set of paths. for example, if $P = \{(1,1), (1,2), (2,1), (2,2)\}$, then $\mathcal{P}(P) = \{\{\}, \{(1,1)\}, \{(1,2)\}, \ldots,\{(1,1),(1,2)\}, \{(1,1),(2,1)\}, \ldots, \{(1,1), (1,2), (2,1)\}, \ldots, \{(1,1), (1,2), (2,1), (2,2)\}\}$. For $\forall x_\theta \in \mathcal{P}(P)$, two sets I_θ and J_θ are defined such that:

$$I_\theta = \{i \in S \mid (i,\bullet) \in x_\theta\} \tag{5.11}$$

and

$$J_\theta = \{ j \in D \mid (\bullet, j) \in x_\theta \}, \tag{5.12}$$

where I_θ and J_θ include the used interfaces in x_θ. Lastly, the optimization problem is to select $x_\theta \in \mathcal{P}(P)$ that minimizes the sum of $\rho(\theta)$ and $u_{t+\Delta t}(\theta)$ given as:

$$Minimize \quad \rho(\theta) + u_{t+\Delta t}(\theta) \tag{5.13}$$

And subject to:

$$u(\theta) \le u_r \wedge |I_\theta| \ge 2 \wedge |J_\theta| \ge 2. \tag{5.14}$$

The last two conditions in (5.14) are used to avoid the single point of failure scenario. In [70], it was indicated that finding the correlation-minimization subset directly is an Integer Quadratic Programming Problem that is NP-hard. Therefore, the subsets fulfilling (5.14) are first arranged in an ascending order based on the sum in (5.13). Then the first subset is selected by RC4CPS.

5.3 Architecture for RC4CPS

RC4CPS approach can be used at the application layer or the transport layer. The approach is not protocol dependent and the main requirement for the approach is the ability to duplicate data packets and to use multiple e2e paths simultaneously. The general architecture of RC4CPS is illustrated in Figure 5.3. The Monitoring & Estimation (M&E) component at the sender monitors the e2e paths in *P* and estimates the attributes of the different subsets of paths from *P*. The collected information is then used by the MP Selection component to select two subsets. These subsets represent the primary and backup subsets of paths, denoted as θ_{pr} and θ_{ba}, that will be used by the end-systems for data transmission. In addition, The M&E component can model each e2e path using one of four MCs to predict future unavailability. The MCs considered in this work are the Gilbert model, the extended Gilbert model, the 3rd-order general Markov model, and the hidden Markov model. A description of these models will be provided in Chapter 7.

The MP selection component uses the information provided by the M&E component and the MP selection formulation presented in Section 5.2 to select θ_{pr} and θ_{ba}. θ_{ba} is used for data transmission only if the behavior of the paths in the θ_{pr} showed abnormality (e.g. all became unavailable). After selecting θ_{pr} and θ_{ba}, the M&E component continues the monitoring and the updating of model parameters of the paths in *P* as well as the estimations of the subsets' attributes. In addition, θ_{pr} is provided to the Data Replicator component to replicate the application data packets over the paths of θ_{pr}.

At the receiver side, the Duplicate Remover will forward the first copy of each packet and discard the rest. The M&E component at the receiver acknowledges the monitoring probes from its counterpart at the sender. Although that the monitoring provides the 2-way unavailability of considered paths which is expected to be different from the

forward and/or backward paths. The required u_r is satisfied for both directions of communication.

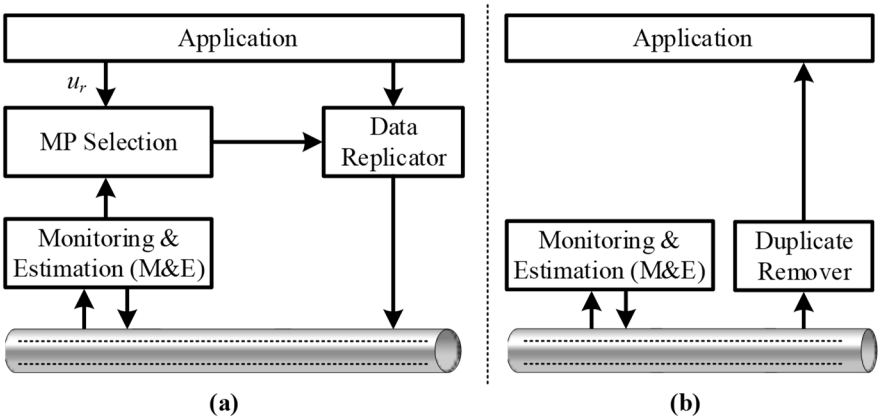


Figure 5.3 Architecture of RC4CPS approach: (a) Sender and (b) Receiver.

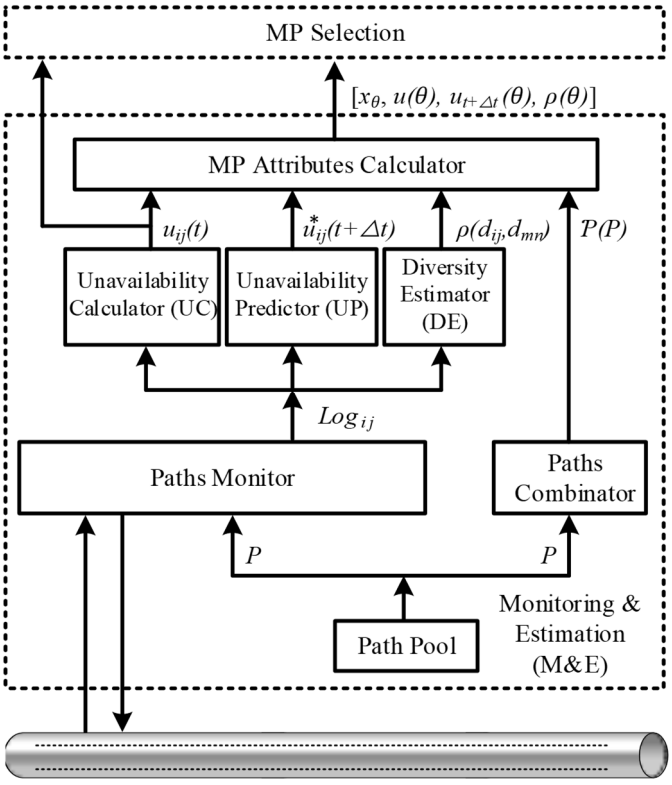


Figure 5.4 Block diagram of the M&E component of RC4CPS.

The block diagram of the M&E component is shown in Figure 5.4 with solid boxes. Here, P is first provided by the Path Pool to the Paths Monitor and Paths Combinator. The Paths Monitor probes all paths in P and updates their logs. The Unavailability Calculator (UC), Unavailability Predictor (UP), & Diversity Estimator (DE) use the paths' logs provided by the Paths Monitor to perform three tasks correspondingly. First, UC approximates $u_{ij}(t)$. Second, UP updates the parameters of the selected Markov model for each path and determines $u^*_{ij}(t+\Delta t)$. The selected model and its initial

parameters for each path are determined in the Initial Monitoring for Model Selection (IMMS) phase (described in Section 5.5). Third, DE estimates the diversity of the different pairs of the monitored paths. The MP Attributes Calculator receives $u_{ij}(t)$, $u^*_{ij}(t+\Delta t)$, and $\rho(d_{ij}, d_{mn})$ for the monitored paths and calculates the attributes matrix (AM) of the form $[x_\theta, u(\theta), u_{t+\Delta t}(\theta), \rho(\theta)]$. The 1st column contains all subsets of P ($x_\theta \in \mathcal{P}(P)$). The 2nd column provides $u(\theta)$ for every x_θ. The 3rd column provides $u_{t+\Delta t}(\theta)$ given in (5.9). Similarly, the 4th column provides $\rho(\theta)$ given in (5.6). Finally, the AM and $u_{ij}(t)$ are forwarded to the MP Selection component to select θ_{pr} and θ_{ba} or exchange them as it will be described in Section 5.4.

5.4 Online Procedures for MP Selection

For the selection process, it is assumed that there is at least one subset that fulfills the selection criteria provided in 5.2.3. The MP selection component first selects θ_{pr}. Then, only the subsets with two more paths other than those used in θ_{pr} are considered when selecting θ_{ba}. Both subsets are selected according to (5.13) and (5.14). If only one subset fulfills the selection criteria, then θ_{ba} remains empty. The MP Selection component forwards either θ_{pr} or θ_{ba} to the Data Replicator. This is determined from $u(\theta)$, concurrent unavailability events on the paths of θ_{pr} and θ_{ba} (when $u_{ij}(t) = 1$ for all paths in a subset), and $u_{t+\Delta t}(\theta)$. These three factors determine also weather an urgent reselection of θ_{pr} and θ_{ba} will be triggered or not. In addition, θ_{pr} and θ_{ba} are periodically reselected to count for variations of path's characteristics.

If both subsets have $u(\theta) < u_r$, then the occurrence of concurrent unavailability events on the paths of θ_{pr} and θ_{ba} during the last transmission of monitoring probes is checked. With such occurrence of concurrent unavailability events, the affected subset is considered to be offline. If θ_{pr} is offline, while θ_{ba} is not, then the MP Selection component switches them, sends the new θ_{pr} to the Data Replicator, and prepones the periodic reselection. In the case that θ_{pr} is offline and θ_{ba} is offline or empty, then an urgent reselection for θ_{pr} and θ_{ba} is triggered. For this urgent reselection, offline subsets are excluded. Moreover, subsets with only one active path (concurrent unavailability events on all paths except one) are also excluded when there are other subsets fulfilling (5.13) and (5.14) with two or more active paths. This will prevent reselecting the same subset again for θ_{pr} and θ_{ba} temporally. This also reduces the communication service unavailability by selecting temporally more available subsets for θ_{pr} and θ_{ba}.

When concurrent events are absent, then $u_{t+\Delta t}(\theta)$ is utilized by the MP Selection component to select between θ_{pr} and θ_{ba}. To avoid frequent exchange between θ_{pr} and θ_{ba} in the presence of a few loses on one of the paths, a counter is utilized. If $u_{t+\Delta t}(\theta_{pr})$ is greater than $u_{t+\Delta t}(\theta_{ba})$ for k sequential transmission, then θ_{pr} and θ_{ba} are exchanged. The motivation for this is to avoid using a subset for data transmission that has a higher probability to become offline as some of its paths start to experience frequent unavailability events.

Lastly, a reselection of θ_{pr} and θ_{ba} is triggered or preponed depending on which subset does not fulfill the u_r bound of maximum allowed unavailability. If $u(\theta)$ for both subsets is greater than u_r, then an urgent reselection is triggered. If only θ_{pr} does not fulfill the u_r bound, then θ_{pr} and θ_{ba} are exchanged and the periodic reselection is preponed.

5.5 Initial Monitoring for Model Selection (IMMS)

the MC model for each e2e path is selected based on the statistics of the path, the achieved accuracy by the model, and the computation complexity of the model which depends on the number of its states [123]–[125]. The IMMS phase is used to determine the initial values of the parameters of the MC model for each e2e path. To achieve this, a binary trace (Section 7.1.1) from the packet trace of the path to be modeled is first generated. An event in the binary trace is designated a 1 and represents an unacknowledged or timed out probe packet in the corresponding packet trace. According to [123], larger sizes of binary traces are preferred in order to accurately model Internet paths. After that, the Cumulative Distribution Functions (CDF) of event-free and event bursts are used to check models accuracy in capturing path statistics. For this purpose, artificial traces are first generated from the given models. Then, the correlation coefficient (*cc*) between the CDFs of event-free bursts and that between the CDFs of event bursts for the artificial and original traces are calculated. *cc* values that are higher than 0.96 represent indications of high accuracy. Further details about model accuracy and selection are provided in Section 7.2.

6 Characterizing Internet Paths Diversity and Unavailability

In this chapter, conducted real world measurements to demonstrate the benefits of MP communication and to answer the 1st research question (Section 2.2) are presented. With this regard, the diversity of different e2e paths and the reduction in communication service unavailability when two and three paths between the source and destination are considered simultaneously are investigated. The chapter describes first the selection of end-systems for the measurements. The data sets collected are described after that. Lastly, the analysis results for the data sets and limitations of the measurements are presented thereafter.

6.1 End-systems Selection

To evaluate the diversity and unavailability of Internet paths, different locations (cities) in Europe were considered. These are *Frankfurt*, *Cologne*, *Stuttgart*, *Warsaw*, *Stockholm*, *Paris*, and *Milan*. In each location two or more nodes that belong to different ISPs were used. This is basically to emulate end-systems with multiple network interfaces but connected to different access networks. In other words, the nodes in each location represent a virtual node (VN) with multiple network interfaces. Where, this VN represents one virtual CPS component. For example, the nodes in *Lemgo*, *Germany*, constitute a multi-interface VN called *Lemgo*. The nodes used were either general purpose *Windows*/*Linux* based computers or ISPs' routers accessed using public Looking Glass (LG) servers. Where, LG servers are web-based portals that provide read-only remote-access to routers of ISPs using one of the publicly available software implementations.

It was not possible to use the PlanetLab platform [126] for the conducted measurements in this chapter. In many cases, there was only one PlanetLab testbed in each city. Even in the case of multiple testbeds in the same city, it was not possible to have at least two testbeds that do not share the same access ISP and/or respond to the ICMP- and TCP-based pings. The other platform considered before starting the measurements is the NorNet platform [127] with multihomed testbeds. However, there was no clear description on how to register and start using the platform.

6.2 Data Sets

The data sets in the conducted evaluations are gathered between the different source-destination pairs using the *Ping* and *Traceroute* tools. In one setup, a more sophisticated versions of the *Ping* tool, namely *hping3* [128], was used. The data set gathered using

the *Traceroute* tool is used to evaluate the diversity while that gathered using the *Ping* tool is used to evaluate the unavailability of the different paths and their combinations.

6.2.1 Traceroute Data Set

For the *Traceroute* data set, the provided tool by *Windows* and *Linux* OSs was used. Each time the tool is launched, a series of ICMP echo packets is sent from the source to the destination with an increasing maximum hop count (Time-To-Live (TTL)) from 1 to 30 and then it stops. The maximum hop count is increased by 1 after each group of three probes. Here, a new probe packet is sent if a reply of the previous probe was received or after a timeout of 5 s.

6.2.2 Ping Data Set

For the Ping probes, two versions of the *Ping* tool were used. The first uses the ICMP protocol while the other, *hping3*, uses the TCP protocol. In this data set, if the source did not receive an ACK within the timeout interval, it is considered that an unavailability event has occurred. As it is expected that a single packet loss might have a significant impact on future CPSs, a stringent timing for the start and end of unavailability events (indicated by unsuccessful transmission of a test packet) was adopted for this data set. That is, the unavailability event starts from the receive time of the reply to the previous successful probe and continues to the send time of the first probe of a series of ten successful probes.

6.3 Diversity Evaluation of Internet Paths

6.3.1 Measurement Setups

For the diversity evaluation, the considered setups are shown in Figure 6.1 (setups in Figures 6.1a and 6.1b will be referred to as *Setup1* and *Setup 2* respectively). The measurements using these setups were conducted at different time intervals, and to different destinations, but using the *Traceroute* tool. The motivation for *Setup 2* is to extend *Setup 1* and include more ISPs from *Europe*. This ensures that initial observations are not limited to a certain country or ISP. The full names of the network acronyms in Figure 6.1 are provided in Table 6.1. The routers in the figure were assigned unique numbers to easily indicate the e2e paths between the different virtual nodes as pairs of the form (*i*,*j*) (see Section 5.1). For example, the e2e path between the *Cogent* router of *Frankfurt* virtual node (router no. 1) and the *Sprint* router of *Milan* virtual node (router no. 17) is indicated by (*1,17)*. Due to the addition and removal of some routers, the assigned numbers are not consecutive. Moreover, in *Setup 1*, a source node and a destination node might share the same number. By contrast, each node (source or destination) in *Setup 2* was assigned a unique number. As shown in Figure 6.1, in each of the considered locations in the setups, two or more routers to emulate multihomed VNs were used.

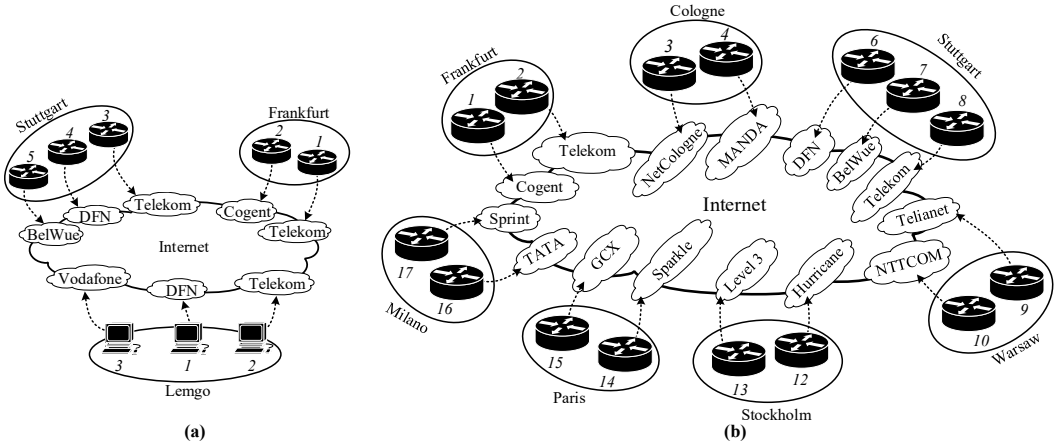


Figure 6.1 The locations and ISPs considered for evaluating the diversity of Internet paths: (a) *Setup 1* and (b) *Setup 2*.

Table 6.1 Full names of networks.

Name/Acronym	Complete Name
AIE	Amsterdam Internet Exchange
AMSIX	Amsterdam Internet Exchange
BelWue	Baden-Wuerttemberg extended LAN
Cogent	Cogent Communications
CW	Cable & Wireless Communications
DE-CIX	German Commercial Internet Exchange
DFN	German National Research and Education Network
ECIX	European Commercial Internet Exchange
EPFL	Swiss Federal Institute of Technology Lausanne
GARR	The Italian Academic & Research Network
GCX	Global Cloud Xchange
GTT	Global Telecom & Technology
Hurricane	Hurricane Electric
Level 3	Level 3 Communications
LINX	London Internet Exchange
MANDA	Metropolitan Area Network Darmstadt
NTTCOM	Nippon Telegraph and Telephone (NTT) Communications
Sparkle	Telecom Italia Sparkle
Sprint	Sprint Corporation
TATA	Tata Communications
Telekom	German Telecom
Telianet	TeliaSonera International Carrier
UniNE	University of Neuchâtel
UPC Austria	United Philips Cable, Austria

Table 6.2 Details of setups for diversity evaluation.

Setup	Start	End
Setup 1 (source nodes 1 and 2)	2015-12-09, 17:30:00	2015-12-22, 14:30:00
Setup 1 (source node 3)	2015-12-10, 11:05:00	2015-12-12, 03:20:00
Setup 2	2016-05-17	2016-06-06

In *Setup 1,* the *Traceroute* sessions were launched from general purpose *Windows* based computers whereas the destination nodes are ISPs' routers that are accessed using public LG servers. In *Setup 2*, the *Traceroute* sessions were launched from the LG servers in each location toward other LG servers in the other locations. As the process of starting a *Traceroute* session was done manually and due to the large number of destinations, I conducted only 3 sessions per destination. The time frame of measurements for both setups is shown in Table 6.2.

6.3.2 Measurement Results

To determine the diversity of considered paths, the IP addresses of hops along each path are mapped to the corresponding ASNs (IP-to-AS mapping) and networks as shown in Tables 6.3 and 6.4. The investigation of path diversity was limited to the traversed networks by the *Traceroute* probes. Therefore, the results of the investigations will be shown at the AS-level rather than at the router level. If the paths do not share any ASNs (networks), then the paths are likely to be disjoint. For the mapping, the regional Internet registries (e.g. [129]) and other databases such as *radb.net* were used. Moreover, many hops in the *Traceroute* traces were associated with DNS names. Information provided by routers DNS names include geographical locations, types of interfaces and their capacities, types and roles of routers, etc. [130]. For example, the DNS name *xr-bie1-ge8-1.x-win.dfn.de* of one of the hops on the path (*1,1*) in *Setup 1* indicates that the router is located in *Bielefeld, Germany*, and belongs to the DFN network.

Some ASNs were found during the mapping to belong to network operators that have merged to/acquisitioned by other larger network operators. Further investigations of these ASNs were done because of the inconsistent information in the regional Internet registries (indicates different operators). As an example, the information provided by the RIPE (abbreviated from French for "European IP Networks") database [129] about AS3209 in Table 6.3 indicated two operators, *ARCOR AG* and *Vodafone*, but *ARCOR AG* was acquisitioned by *Vodafone* in 2009. Therefore the ASN is considered to belong to *Vodafone* in the Network column in Table 6.3. There is also a partnership between *GTT* and *GCX* according to the information available in the Internet which explains why *GTT* was the first network in some *Traceroute* sessions originating from *GCX* routers in *Setup 2*.

Due to the large amount of results' data and for clearer description of the results, I present only the diversity results for *Setup 1* (Table 6.3) and the diversity results between two VNs in *Setup 2* (*Frankfurt* and *Milan*). The complete diversity results for *Setup 2* are provided in Appendix A. At the end of this section, the results from both setups are summarized.

Table 6.3 Diversity of considered Internet paths in S*etup 1*.

Path	# of Hops	# of Sessions[1]	Path Persistency (%)	ASNs[1,2]	Networks
(1,2)	13	1237	99.51	AS680, AS174	DFN, Cogent
(1,3)	8	1237	99.92	AS680, AS3320	DFN, Telekom
(1,4)	9	1237	99.92	AS680	DFN
(1,5)	11	1237	99.92	AS680, AS553	DFN, BelWue
(2,1)	5	1236	99.11	AS3320	Telekom
(2,2)	7	1236	99.92	AS3320, AS174	Telekom, Cogent
(2,3)	5	1236	100	AS3320	Telekom
(2,5)	11	1236	100	AS3320, AS1299, AS553	Telekom, TeliaSonera AB, BelWue
(3,1)	9	161	100	AS3209, AS3320	Vodafone, Telekom
(3,2)	14	161	91.92	AS3209, AS1273, AS174	Vodafone, Cogent
(3,3)	9	161	100	AS3209, AS3320	Vodafone, Telekom
(3,5)	11	161	100	AS3209, AS51531, AS553	Vodafone GmbH, DE-CIX, BelWue

[1.] In the direction from sources to destinations.
[2.] Probes from destinations to sources traversed the same networks but in reverse direction.

Table 6.4 Diversity of considered Internet paths between *Frankfurt* and *Milan* VNs in S*etup 2*.

Path	Destination reached?	# of Hops	ASs	Networks
(1,16)	Y	6	AS174, AS6453	Cogent, TATA
(1,17)	Y	7	AS174, AS1239	Cogent, Sprint
(2,16)	Y	6	AS3320, AS6453	Telekom, TATA
(2,17)	Y	6	AS3320, AS1239	Telekom, Sprint

As it can be seen from Table 6.3, three pairs of disjoint paths are available in *Setup 1* for the *Lemgo-Frankfurt* VNs pair. Namely {(*1*,*2*), (*2*,*1*)}, {(*1*,*2*), (*3*,*1*)}, and {(*2*,*1*), (*3*,*2*)}. Such paths that traverse completely different networks provide more communication reliability, especially when large network outages happen (such as those indicated in [7]). Table 6.3 also provides the persistency of each path as the percentage of *Traceroute* sessions for which the path did not change. It is necessary to indicate here that hop changes caused by load balanced links between routers were not considered as path changes. Load balanced links were identified by the presence of one hop with two IP addresses but identical DNS name between two unvarying hops. Similar path pairs can also be identified in *Setup 1* for the *Lemgo-Stuttgart* pair of VNs.

In the column *Destination Reached?* of Table 6.4 as well as Tables A.13- A.27 , the last

Table 6.5 Possible disjoint 2-path subsets between the different VNs in S*etup 2*.

Network (Nodes)	**Frankfurt (1,2)**	**Cologne (3,4)**	**Stuttgart (6,7,8)**	**Warsaw (9,10)**	**Stockholm (12,13)**	**Paris (14,15)**	**Milan (16,17)**
Frankfurt (1,2)	x	{(1,3),(2,4)}, {(1,4),(2,3)}	{(1,7),(2,8)}	{(1,9),(2,10)}, {(1,10),(2,9)}	{(1,12),(2,13)}, {(1,13),(2,12)}	{(1,14),(2,15)}, {(1,15),(2,14)}	{(1,16),(2,17)}, {(1,17),(2,16)}
Cologne (3,4)	{(3,1),(4,2)}, {(3,2),(4,1)}	x	{(3,7),(4,8)}, {(3,8),(4,7)}	{(3,9),(4,10)}, {(3,10),(4,9)}	{(3,12),(4,13)}, {(3,13),(4,12)}	{(3,14),(4,15)}, {(3,15),(4,14)}	{(3,16),(4,17)}, {(3,17),(4,16)}
Stuttgart (6,7,8)	{(7,1),(8,2)}, {(6,1),(7,2)}	{(7,3),(8,4)}, {(7,4),(8,3)}	x	{(6,9),(7,8)}, {(6,10),(7,9)}	{(7,12),(8,13)}, {(7,13),(8,12)}	{(7,14),(8,15)}, {(7,15),(8,14)}	{(7,16),(8,17)}, {(7,17),(8,16)}
Warsaw (9,10)	{(9,1),(10,2)}, {(9,2),(10,1)}	{(9,3),(10,4)}, {(9,4),(10,3)}	{(9,7),(10,8)}, {(9,8),(10,7)}	x	{(9,12),(10,13)}	{(9,14),(10,15)}, {(9,15),(10,14)}	{(9,16),(10,17)}, {(9,17),(10,16)}
Stockholm (12,13)	{(12,1),(13,2)}, {(12,2),(13,1)}	{(13,3),(12,4)}, {(13,4),(12,3)}	{(13,7),(12,8)}, {(13,8),(12,7)}	{(12,9),(13,10)}	x	{(13,14),(12,15)}, {(13,15),(12,14)}	{(13,16),(12,17)}, {(13,17),(12,16)}
Paris (14,15)	{(14,1),(15,2)}, {(14,2),(15,1)}	{(14,4),(15,3)}	{(14,7),(15,8)}, {(14,8),(15,7)}	{(14,9),(15,10)}, {(14,10),(15,9)}	{(14,13),(15,12)}, {(14,12),(15,13)}	x	{(14,16),(15,17)}, {(14,17),(15,16)}
Milan (16,17)	{(16,1),(17,2)}, {(16,2),(17,1)}	{(16,3),(17,4)}, {(16,4),(17,3)}	{(16,7),(17,8)}, {(16,8),(17,7)}	{(16,9),(17,10)}, {(16,10),(17,9)}	{(16,13),(17,12)}, {(16,12),(17,13)}	{(16,14),(17,15)}, {(16,15),(17,14)}	x

hop IP and the traced IP are compared. "*Y"* indicates that the destination was reached. "*Net"* indicates that one or more hops in the network of the destination responded, but not the destination. "*!Net"* indicates that none of the routers (hops) in the network of the destination responded (which, as mentioned in Section 6.5, might be attribute to the drop of ICMP traffic by boarder routers between networks). This is also indicated in the last two columns of the tables using asterisks, where no further information about the networks along the path could be extracted. In these two columns, the traversed ASNs and networks by each e2e path between the different source-destination pairs are listed. In the case where an IP address has more than one ASN, I reported both in the tables but separated with '/'. The last hop IP in some *Traceroute* sessions in *Setup 2* was not the same as the traced IP, even though that the DNS names agree. For these sessions, the *Ally* tool [131] was used to check if both IPs belong to the same router or not.

As provided in Table 6.4, two options of 2-path subsets exist where the paths traverse completely different networks. To present the results in Table 6.4 and those in the Appendix A in a simpler way, Table 6.5 is provided. In the table, the VNs and the corresponding routers numbers are listed. As mentioned in Chapter 5, with multiple e2e paths between a source and a destination, different subsets of these paths are available. To indicate the diversity of e2e paths between the different VNs, Table 6.5 lists up to two possible subsets of two paths. The paths in each of these subsets traversed completely different networks. Other subsets with different number of e2e paths might also be possible, but they are not provided in the table. As an example, the two subsets of e2e paths between *Frankfurt* and *Milan* are {(*1*,*16*), (*2*,*17*)} and {(*1*,*17*), (*2*,*16*)}. In the case of *Frankfurt* to *Stuttgart*, there was only one possible subset of two e2e paths.

The results presented in this section show clearly that e2e paths traversing completely different networks (which are likely to be disjoint) can be attained using different access ISPs.

6.4 Unavailability Evaluation of Internet Paths

6.4.1 Measurement Setups

For the unavailability evaluation, the two considered setups are shown in Figure 6.2. These setups were used at separate time intervals, to different destinations, and using different tools. This is mainly to ensure that the measurements are not biased by, for example, the tool being used to conduct the measurements. In both setups, one group of source nodes located in *Lemgo* and two or more destination groups located in other cities were used. In addition, all source nodes are time synchronized to the same time servers and used wired connections. For the networks presented in the figure with acronyms, the complete names are in Table 6.1.

In order to facilitate the analysis of the results, the sources and destinations in both setups were also numbered in a similar fashion to that in Figure 6.1. The details of how

the numbers for the different nodes were assigned are described in Section 6.3.1. The used tools and duration of measurement for both setups are provided in Table 6.6. As the target here is to evaluate unavailability rather than implementing concurrent duplicate transmission, the test packets departure times from the different sources are not synchronized in these measurements.

Table 6.6 Details of setups for unavailability evaluation.

Setup	Tool	Start	End
Setup 1	Ping (ICMP)	2015-12-09, 17:30:00	2015-12-22, 14:30:00
Setup 2	hping3 (TCP)	2016-06-21, 00:00:00	2016-06-28, 06:00:00

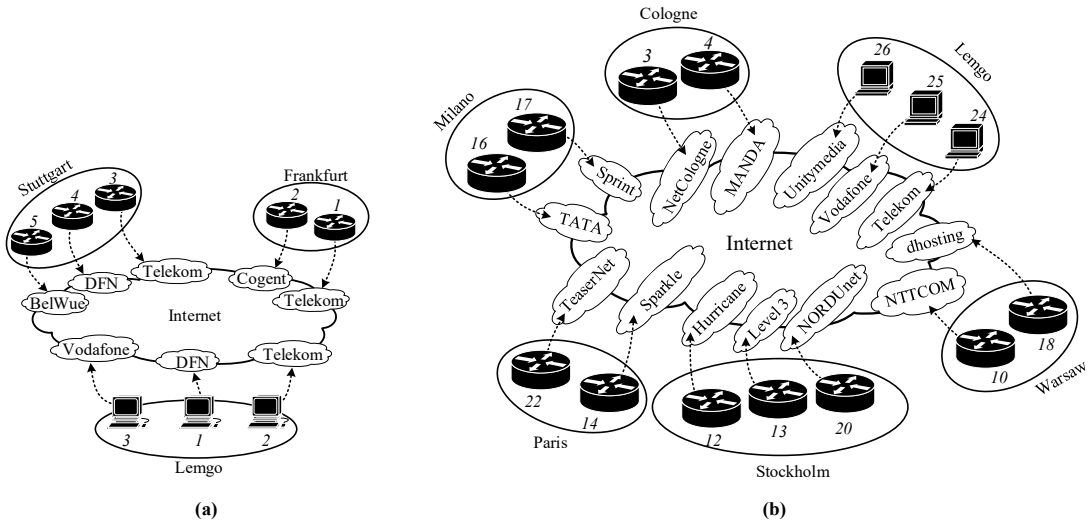


Figure 6.2 Location and ISPs considered for evaluating the unavailability of Internet paths: (a) *Setup 1* and (b) *Setup2*.

For the first setup, the considered locations, the used access ISPs, and the assigned numbers to the different nodes are illustrated in Figure 6.2a. The sources constituting the *Lemgo* source VN used ICMP echo request packets with 100 bytes of payload and probed the different paths to the *Frankfurt* and *Stuttgart* destination VNs. In this setup, a similar probe frequency to that proposed in [11] is used. As shown in Figure 6.3, each e2e path is probed every 15 s. If the source did not receive a reply within 3 s (timeout interval), it is considered that an unavailability event has occurred and the probe frequency is increased to be every 5 s. The end of unavailability event (see Section 6.2.2) also restores the probe frequency to be every 15 s. Moreover, it was not possible to conduct the measurements between each source and all destinations in *Setup 1* which might be attributed to the way that different networks treat the probe packets, namely ICMP packets, heading to or coming from a different network [11].

In the second setup, a larger number of locations were considered as shown in Figure 6.2b. In addition, the *hping3* tool was used which can send customized TCP/IP packets and provide the results in a similar way to that of the *Ping* tool in famous OSs (e.g.

Linux). In this setup, the sources located in *Lemgo* probe all e2e paths to the different destinations every 5 s with timeout of 1 s (default timeout of the tool). The start and end of unavailability events are determined in a similar fashion to that in *Setup 1*. However, the occurrence of an unavailability event does not influence the probing frequency.

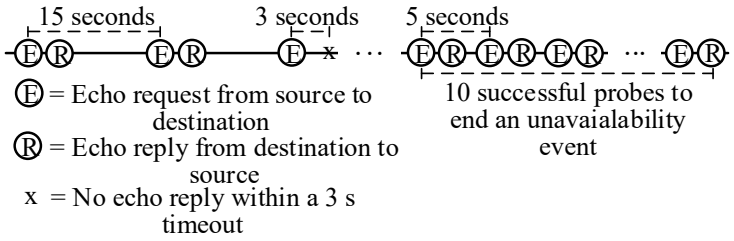


Figure 6.3 Frequency of the *Ping* probes in *Setup 1*.

6.4.2 Measurement Results

For convenience of view in this section, only the detailed unavailability results from *Setup 1* and that between *Lemgo* VN and one of the destination VNs from *Setup 2*. The complete unavailability results to the remaining locations in *Setup 2* are provided in Appendix A. After that, the results from both unavailability evaluation setups are summarized at the end of this section.

As the send and receive times of probe packets are logged (for timed out probes, only the send time is logged beside an indication of time out) in the measurements, the instantaneous unavailability function $u_{ij}(t)$ (given in (5.1)) for each path over the measurement interval T is available. Then, for each 1-, 2-, and 3-path subset, the results of (5.4) was multiplied by 100 and by T to obtain each subset's approximate unavailability in terms of the percentage of measurement interval T ($u(\theta)$ (%)) and in terms of the total number of seconds ($u(\theta)$ (s)) correspondingly.

The unavailability results for each 1-, 2-, and 3-path subset to both destination VNs *Frankfurt* and *Stuttgart* in *Setup 1* are presented in Tables 6.7 - 6.9. As illustrated in Table 6.7, $u(\theta)$ (which equals u_{ij} when $|\theta|$ equals 1) of most e2e paths falls below 1%. In case of path (*2*,*1*), a high percentage of unavailability of about 15% is observed. Such value might be attributed to the low priority assigned to ICMP packets by routers along the path in cases of traffic increase and congestion. Nevertheless, this high percentage is considered as an indication of high load and/or frequent short congestions over the path. In Table 6.8, the $u(\theta)$ of the different possible subsets of 2 paths to each destination VN in *Setup 1* is listed. For example, $u(\theta)$ of the 2-path subset {(*1*,*2*), (*2*,*1*)} is obtained as:

$$u(\theta) = \frac{1}{T}\int_0^T \left(u_{(1,2)}(t) \quad u_{(2,1)}(t) \right) dt. \tag{6.1}$$

Here, $u(\theta)$ of almost all subsets of e2e paths that do not share ASs is 0 as in the case of the subset {(*1*,*2*), (*2*,*1*)}. In addition, all other subsets have $u(\theta)$ of less than 0.1% even when used with the path (*2*,*1*) with unavailability of about 15%. This indicates that the

concurrency of unavailability events on the different paths is very small. *u(θ)* of

Table 6.7 Unavailability results for each e2e path in *Setup 1*.

Path	$u(\theta)$ (%)	$u(\theta)$ (s)
(1,2)	0.0634	704.91
(1,3)	0.0965	1073.71
(1,4)	0.0296	329.19
(1,5)	0.0671	746.74
(2,1)	15.2879	170062.88
(2,2)	0.2657	2955.38
(2,3)	0.0487	541.68
(2,5)	1.0734	11940.64

Table 6.8 Unavailability results for the 2-path subsets in *Setup 1*.

Destination	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
Frankfurt	*{(1,2),(2,1)}*	0.0024	26.6
	{(1,2),{(2,2)}	0.0000	0
	{(2,1),(2,2)}	0.0437	486.6
Stuttgart	*{(1,3),(1,4)}*	0.0108	119.7
	{(1,3),(1,5)}	0.0239	266.2
	{(1,3),(2,3)}	0.0081	89.8
	{(1,3),(2,5)}	0.0012	12.9
	{(1,4),(1,5)}	0.0082	91.1
	{(1,4),(2,3)}	0.0000	0.0
	{(1,4),(2,5)}	0.0000	0.0
	{(1,5),(2,3)}	0.0000	0.0
	{(1,5),(2,5)}	0.0000	0.0
	{(2,3),(2,5)}	0.0034	38.0

Table 6.9 Unavailability results for the 3-path subsets in *Setup 1*.

Destination	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
Frankfurt	*{(1,2),(2,1),(2,2)}*	0.0000	0
Stuttgart	*{(1,3),(1,4),(1,5)}*	0.0082	91.1
	{(1,3),(1,4),(2,3)}	0.0000	0
	{(1,3),(1,4),(2,5)}	0.0000	0
	{(1,3),(1,5),(2,3)}	0.0000	0
	{(1,3),(1,5),(2,5)}	0.0000	0
	{(1,3),(2,3),(2,5)}	0.0000	0
	{(1,4),(1,5),(2,3)}	0.0000	0
	{(1,4),(1,5),(2,5)}	0.0000	0
	{(1,4),(2,3),(2,5)}	0.0000	0
	{(1,5),(2,3),(2,5)}	0.0000	0

3-path subsets are provided in Table 6.9. From the table, 3-path subsets show even better results with *u(θ)* equal to 0 for all subsets except the subset of paths from source node *1* (of the *Lemgo* VN) to the destinations in *Stuttgart*. This is mainly attributed to the first shared hops for all of these paths in the AS680. The accumulative number of unavailability events along with the average unavailability duration for all 1-, 2-, and 3-path subsets is shown in Figure 6.4. The figure indicates also the destination VNs and the number of combined paths. For example, *F: 1 path* refers to 1-path subsets to

Frankfurt VN. Even though that all subsets of paths show comparable average duration of unavailability events, the numbers of unavailability events for 1-path subsets are the highest.

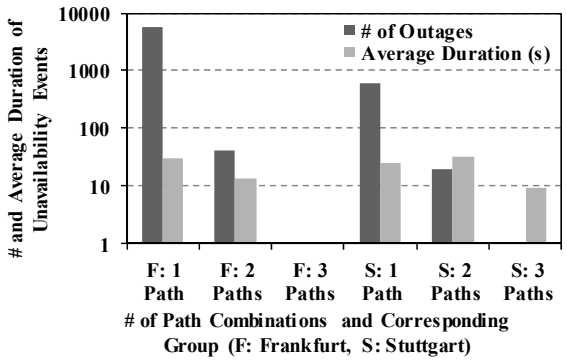


Figure 6.4 Number and average duration of unavailability events for the different subsets of e2e paths in *Setup 1* for unavailability evaluation.

Table 6.10 Unavailability of e2e paths between *Lemgo* and *Paris* VNs in *Setup 2*.

Path	$u(\theta)$ (%)	$u(\theta)$ (s)
(24,14)	0.5444	3410.353
(24,22)	0.9347	5854.979
(25,14)	0.2846	1782.678
(25,22)	0.2448	1533.460
(26,14)	0.7864	4926.085
(26,22)	0.1022	640.335

For *Setup 2*, only the detailed unavailability results between *Lemgo* and *Paris* VNs are reported. As it can be seen from Table 6.10 and similar to *Setup 1* results, unavailability of all e2e paths were below 1 %. In addition, the unavailability of 2- and 3-path subsets between *Lemgo* and *Paris* VN are provided in Tables 6.11 and 6.12.

In the following, the results obtained from both measurement setups are summarized separately. This is because each setup used a different tool and/or has a different probing frequency and measurement's interval.

The unavailability of probed e2e paths in *Setup 1* as provided in Table 6.7 was in the range 0.0296-15.2879% of the measurement interval. On the other hand, the unavailability of 2- and 3-path subsets as provided in Tables 6.8 and 6.9 was in the range 0-0.0437% and 0-0.0082% of the measurement interval correspondingly. In the case of *Setup 2*, the measured unavailability of e2e paths between *Lemgo* VN and the different destination VNs was between 0.1022% and 1.1272% of the measurement period. In contrast, the 2- and 3-path subsets have unavailability of 0-0.0012% of the measurement interval.

Table 6.11 Unavailability of the 2-path subsets between *Lemgo* and *Paris* VNs in *Setup 2*.

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,14),(24,22)}	0.0007	436.912
{(24,14),(25,14)}	0.0000	0.000
{(24,14),(25,22)}	0.0000	0.000
{(24,14),(26,14)}	0.0001	34.662
{(24,14),(26,22)}	0.0000	0.000
{(24,22),(25,14)}	0.0000	19.659
{(24,22),(25,22)}	0.0000	20.069
{(24,22),(26,14)}	0.0001	39.278
{(24,22),(26,22)}	0.0000	2.350
{(25,14),(25,22)}	0.0010	643.384
{(25,14),(26,14)}	0.0000	17.367
{(25,14),(26,22)}	0.0000	0.000
{(25,22),(26,14)}	0.0000	13.413
{(25,22),(26,22)}	0.0000	2.211
{(26,14),(26,22)}	0.0000	0.000

Table 6.12 Unavailability of the 3-path subsets between *Lemgo* and *Paris* VNs in *Setup 2*.

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,14),(24,22),(25,14)}	0.0000	0.000
{(24,14),(24,22),(25,22)}	0.0000	0.000
{(24,14),(24,22),(26,14)}	0.0000	0.000
{(24,14),(24,22),(26,22)}	0.0000	0.000
{(24,14),(25,14),(25,22)}	0.0000	0.000
{(24,14),(25,14),(26,14)}	0.0000	0.000
{(24,14),(25,14),(26,22)}	0.0000	0.000
{(24,14),(25,22),(26,14)}	0.0000	0.000
{(24,14),(25,22),(26,22)}	0.0000	0.000
{(24,14),(26,14),(26,22)}	0.0000	0.000
{(24,22),(25,14),(25,22)}	0.0000	9.563
{(24,22),(25,14),(26,14)}	0.0000	0.000
{(24,22),(25,14),(26,22)}	0.0000	0.000
{(24,22),(25,22),(26,14)}	0.0000	0.000
{(24,22),(25,22),(26,22)}	0.0000	0.000
{(24,22),(26,14),(26,22)}	0.0000	0.000
{(25,14),(25,22),(26,14)}	0.0000	10.019
{(25,14),(25,22),(26,22)}	0.0000	0.000
{(25,14),(26,14),(26,22)}	0.0000	0.000
{(25,22),(26,14),(26,22)}	0.0000	0.000

It is clear from the results that MP communication over Internet can support the high communication availability required by many CPSs that span large geographical areas such as smart grids. For such CPSs, the communication service unavailability was required to be between 1% and 0.00001% [13]. In addition, and with regard to the unavailability of different e2e paths, the results in this chapter agree with the results in previous works in the literature [11], [48].

6.5 Limitations of Measurements

In three of the setups used to evaluate diversity and unavailability, ICMP-based tools were used. For these setups, it was taken into account that ICMP packets might be treated with lower priority by some routers in the Internet [11]. Nevertheless, the use of ICMP packets in this case can still reveal the potential of MP communication over the Internet. In the diversity evaluation, the interest was in identifying the networks traversed by each e2e path. Therefore, the use of ICMP-based *Traceroute* is enough as long as the destinations are reached and the IP addresses of the hops can be identified. Similarly, the use of ICMP-based *Ping* is suitable in this case because if MP communication can reduce unavailability of communication service when using not prioritized traffic, then it is expected to get even better results when using prioritized one. In addition, in *Setup 2* for the unavailability evaluation, a TCP-based *Ping* tool was used and similar results (except in one case) as those obtained using ICMP were observed.

Another issue is the number of *Traceroute* sessions for the diversity evaluation. As the destinations in *Setup 1* are ISPs' routers with no possibility to run scripts from them, only a few *Traceroute* sessions from destinations to sources were carried out compared to that from sources to destinations. Likewise, the number of *Traceroute* sessions for each e2e path in *Setup 2* was also small due to the same reason. However, in both setups, *Traceroute* sessions from destinations to sources traversed the same networks as those from sources to destinations. It is also expected that the *Ping* probes and their corresponding replies between the different source-destination pairs traversed the same networks traversed by the *Traceroute* sessions as the routing is usually based on destination networks.

In the diversity evaluation, tools such as *Paris Traceroute* [132] might yield better results with regard to finding load balanced links and all possible paths between the source and the destination. However, the use of such tools in this study is not necessarily as the legacy *Traceroute* tool can capture the IP addresses of hops over the possible paths and the IP-to-AS mapping will return the corresponding networks (only mapping to the AS-level is needed).

7 Online Monitoring and Prediction

In this chapter, the mechanisms used in the M&E component of RC4CPS approach to carry out the required tasks are described. The M&E component and its architecture were briefly introduced in Chapter 5. For clearness of information presentation in this chapter, I reintroduce the M&E component in Figure 7.1. As it can be seen in the figure, the main functionality of this component is centered on determining unavailability, predicting future unavailability, and estimating diversity of the different path subsets of P that represent the set of all e2e paths.

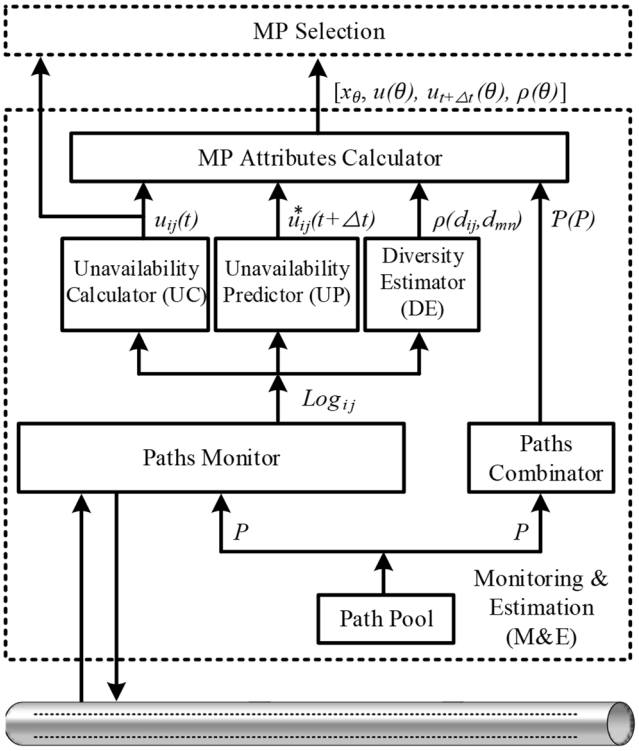


Figure 7.1 Architecture of M&E component.

This chapter first introduces the possible models considered in this work to model an e2e path. After that, the accuracy of a given model in capturing the e2e behavior of a path is investigated. The impacts of probing packets frequency and type on the approximated average unavailability will be also described. Lastly, an example about diversity estimation using the comparison test (Section 5.2.1) is provided.

7.1 Modeling e2e Path Unavailability Using Markov Chains

MCs are widely used in the literature to model Internet e2e paths (e.g. [124], [125], [133]–[136]). This is attributed to the structured nature of these chains that facilitate the analysis of the temporal dependencies of random processes. In other words, in such

class of random processes, the process has a finite-number of states where the current state is determined by the history of the process. Another approach to model the unavailability of e2e paths used an equation to model the time to repair cumulative distribution function and assumed independent and exponentially distributed inter-arrival times of unavailability events [48]. As indicated by the authors of the work, it was difficult to determine an upper bound for the durations of availability events and, consequently, to characterize their distribution. The unavailability probability in such model is expected to be the probability of having an unavailability event with a duration that will impact the next transmission. However, it is not clear if the current state of the path (available/unavailable) will impact the prediction of an unavailability event and how the equation used will be updated when the characteristic of the Internet path change. With this regard, MCs are more suitable for online (on the run) unavailability prediction. For example, the TM in MCs can be updated after each transmission and therefore can reflect changes in the characteristics of the path (e.g. whether the number of unavailability events is increasing or decreasing). Based on the current state, the probabilities of the possible next states (available/unavailable) can be determined. As a result, the current state of the path impacts the probability of an unavailability event.

7.1.1 Path Traces

To model the different e2e paths, binary traces are used, where a binary trace consists of sequences of 0s and 1s. The binary trace for an e2e path is obtained from the corresponding packet trace of the path. An unacknowledged or timed out probe packet in a packet trace is designated a 1 while an acknowledged probe packet is designated a 0. The reason for such conversion rather than using, for example, sequence numbers is to facilitate the modeling of Internet paths using MCs. This is mainly because the occurrence of an event is more important than, for example, its reason or type for RC4CPS. An event in this context can be, for example, a lost packet, a packet received with error or a packet with time delay higher than the maximum threshold.

7.1.2 Gilbert Model

The Gilbert model is one of the simple MCs used to model the e2e characteristics of Internet paths. The model was proposed by Gilbert [137] and consists of two states only. The first state is the GOOD state that produces no event or 0s, while the other state is the BAD state that produces the events or 1s. More specifically, each state in a Gilbert model represents one of the symbols (0 or 1) in the binary trace of events. This trace is defined as $\{Z_i\}_{i=1}^{n}$ where n is the length of the binary trace and $Z_i \in \{0,1\}$ is the binary random variable for event or non-event resulted from the i^{th} packet transmission. As these binary traces represent packet traces, the transition between the Gilbert model states are per packet and depend only on the current state.

The parameters of the Gilbert model, as shown in Figure 7.2, are p_{01} and p_{10}. p_{01} is the probability that the next packet transmission will result in an event (when the packet is lost or timed out) given that the last packet was acknowledged. p_{10} is the probability that the next packet transmission will result in no event (when the packet's ACK is

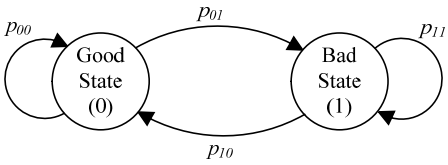


Figure 7.2 The Gilbert model.

received), given that the last packet was not acknowledged. The parameters of the Gilbert model are often represented in a matrix form given by:

$$\begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix}. \tag{7.1}$$

These parameters can be estimated using the event burst's length distribution statistics as proposed in [135]. According to this approach, first m_i is defined as the number of event burst (unacknowledged packets in this case and represented by sequences of 1s in the binary trace) of length i, where $i \in \{1, 2, \ldots, n\text{-}1\}$ and $n - 1$ is the length of the longest event burst in the binary trace. Then m_0 is defined as the number of packets that resulted in no event and represented as 0s in the binary trace. After that, p_{01} and p_{10} of the Gilbert model are calculated as follows:

$$p_{01} = \left(\sum_{i=1}^{n-1} m_i \right) \Big/ m_0 \tag{7.2}$$

and

$$p_{10} = 1 - \left(\sum_{i=2}^{n-1} m_i \cdot (i-1) \right) \Big/ \left(\sum_{i=1}^{n-1} m_i \cdot i \right). \tag{7.3}$$

In addition, the probabilities for the chain to be in one of the two states (state probabilities) can be computed as follows:

$$\pi_0 = \frac{p_{10}}{p_{01} + p_{10}} \tag{7.4}$$

and

$$\pi_1 = \frac{p_{01}}{p_{01} + p_{10}} \tag{7.5}$$

Even though that the Gilbert model is simple to understand and easy to implement, the model is memoryless and can't capture bursty behavior of events.

7.1.3 Extended Gilbert Model

An extension to the Gilbert model to deal with bursty behavior is the Extended Gilbert model (EGM) which was proposed in [135]. The model uses n+1 states to remember n

previous events as shown in Figure 7.3. The main difference between the EGM and a general Markov model is the number of states needed to remember n previous values and when to remember them. More specifically, to remember n previous values, the general Markov model needs 2^n states compared to only n+1 for the EGM. This is mainly because in the general Markov model all past n values (whether they indicate occurrence of events or not) contribute to the future value. In the EGM, by contrast, the past up to n values that indicate events contribute to the future value. This significant reduction in states reduces also the implementation and computation complexity of the model. Nevertheless, the model does not specifically capture the burstiness of 0s (non-event sequences representing inter-event distances consisting of 0s).

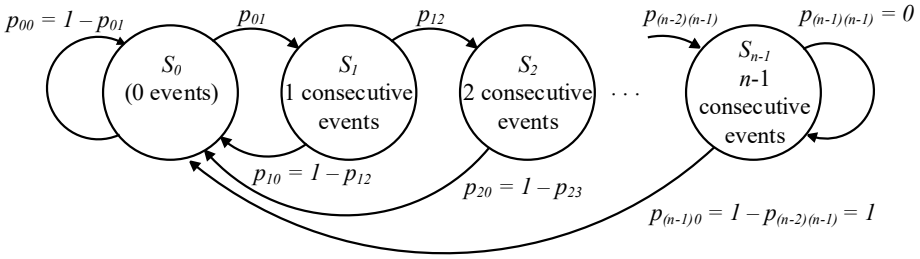


Figure 7.3 The extended Gilbert model.

As it can be seen in Figure 7.3, each state i, where i = 0, 1, …, n-2, indicates the number of events occurred since the beginning of the current events burst while state n-1 indicates that n-1 or more events have occurred. The parameters to be determined for the model are the transition probabilities between the successive states, that is $p_{i(i+1)}$. These parameters can be represented in a matrix form as follows:

$$\begin{bmatrix} p_{00} & p_{10} & p_{20} & \cdots & p_{(n-2)0} & p_{(n-1)0} \\ p_{01} & 0 & 0 & \cdots & 0 & 0 \\ 0 & p_{12} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p_{(n-2)(n-1)} & p_{(n-1)(n-1)} \end{bmatrix}. \tag{7.6}$$

The equations to calculate these parameters as given in [135] are as follows:

$$p_{01} = \left(\sum_{i=1}^{n-1} m_i \right) \Big/ m_0 \tag{7.7}$$

and

$$p_{(k-1)k} = 1 - \left(\sum_{i=k}^{n-1} m_i \right) \Big/ \left(\sum_{i=k-1}^{n-1} m_i \right). \tag{7.8}$$

Beside the model parameters, the n-1 value (state) needs to be determined in order to be used in the model. For binary traces with short event bursts, n-1 can be selected to be the longest event burst. However, and as it has been observed from the traces captured in the measurements (Section 6.4), some traces might experience some event bursts of 20 consecutive events. In this case, selecting n-1 value to equal the longest event burst might inaccurately model the trace, especially when these burst lengths occur at a very

low frequency. Therefore, and as suggested in [133], *n*-1 for the EGM is selected to equal the event burst length with a percentile of 99th or more. As shown in Figure 7.4, the event bursts of the e2e paths (*2*,*1*) and (*2*,*2*) from *Setup 1* for unavailability evaluation (Section 6.4) with greater than or equal to 99th percentile are 8 and 2 correspondingly.

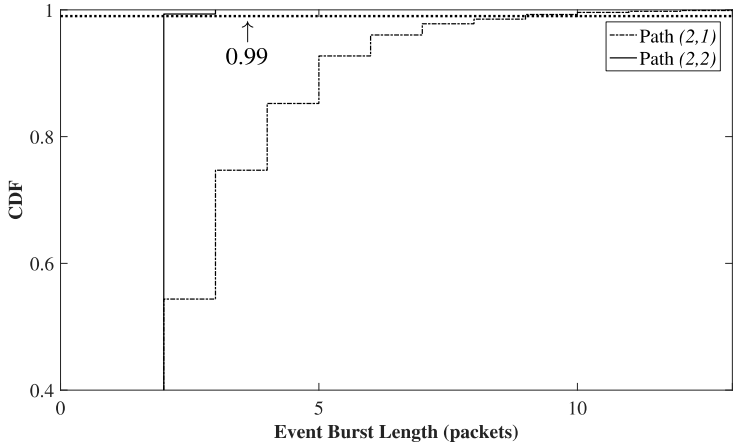


Figure 7.4 Cumulative distribution of event bursts for the e2e paths (*2*,*1*) and (*2*,*2*) from *Setup 1* for unavailability evaluation (Section 6.4).

7.1.4 General Markov Chain Model

A more general model than the previous two models is the n^{th}-order MC with 2^n states that are defined by the set *S*. If Z_i is defined as in Section 7.1.2, then, the parameters to be determined for the model are the transition probabilities $Pr[Z_i = z_i \mid Z_{i-1} = z_{i-1}, Z_{i-2} = z_{i-2}, \ldots, Z_{i-n} = z_{i-n}]$ for all $Z_i, Z_{i-1}, Z_{i-2}, \ldots, Z_{i-n}$ combinations. It is clear here that the value of the next binary random variable Z_i depends on the last *n* values and is generated when the chain is in the state $z_{i-1}, z_{i-2}, \ldots, z_{i-n} = s \in S$.

For the estimation of the parameters of the n^{th}-order MC, $z = (z_1, z_2, \ldots, z_k)$ is first defined to be a binary trace of events obtained from a Markov source. After that, the transition probabilities of the MC as provided in [125] for all $l \in \{0,1\}$ and $s \in S$ are determined. For this, $C_z^n(l,s)$ is defined as the count of times where state *s* is followed by *l* and is given as:

$$C_z^n(l,s) = \sum_{i=1}^{k} \mathrm{I}\{z_i = l, (z_{i-1}, \ldots, z_{i-n}) = s\}, \tag{7.9}$$

where I(·) is the indicator function with a value of 1 for every matching binary sequence to $\{z_i = l, (z_{i-1}, z_{i-2}, \ldots, z_{i-n}) = s\}$ and 0 otherwise.

In addition, $C_z^n(s)$ is defined as the count of times where state *s* is seen and is given as:

$$C_z^n(s) = \sum_{i=1}^{k} \mathrm{I}\{(z_{i-1}, \ldots, z_{i-n}) = s\}. \tag{7.10}$$

Lastly, if $w_z^n(l|s)$ denoted the probability that the next value z_i will be *l* given that the current state is $s = (z_{i-1}, z_{i-2}, \ldots, z_{i-n})$, where $s \in S$, then $w_z^n(l|s)$ is an estimate of the

transition probability from state s to state (l, z_1, …, z_{n-1}). Hence, the estimate of the transition probabilities of an n^{th}-order MC can be obtained as:

$$w_z^n(l \mid s) = \begin{cases} \dfrac{C_z^n(l,s)}{C_z^n(s)} & if\ C_z^n(s) > 0 \\ 0 & otherwise \end{cases} \tag{7.11}$$

An important issue to be addressed here is the order to be selected for the MC. More specifically, how many past values can influence the next value of the chain. This problem has been addressed in [138] and [139], where the n^{th}-order conditional entropy was used to determine the proper order of the chain. This approach was used by the authors in [125] where they have indicated that entropies of MCs of order higher than 3 were almost constant. Therefore, only general MCs of 3rd order are considered in this work to model the binary traces of the different e2e paths.

7.1.5 Hidden Markov Chain Model

Another Markov model with well-known use in pattern recognition applications is the Hidden Markov Model (HMM). Here, each pattern is associated with a hidden state.

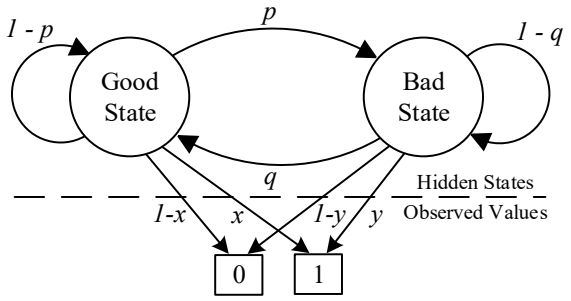


Figure 7.5 The Hidden Markov model with two states.

This allows the modeling of data patterns with different characteristics (i.e. patterns of events). Similar to the previous simpler Markov models, the parameters of the model to be determined are the transition probabilities between the states and the memory of the chain. In addition, and as shown in Figure 7.5, the conditional probabilities of event or no event occurrences for each state are also needed for HMMs. These parameters can be represented using matrices as follows:

$$TM = \begin{bmatrix} (1-p) & p \\ q & (1-q) \end{bmatrix} \tag{7.12}$$

and

$$E = \begin{bmatrix} x & y \end{bmatrix}, \tag{7.13}$$

where TM is the transition probabilities matrix between the hidden states and E is the emission matrix for each state. From (7.12), (7.13), and Figure 5.4, it can be observed that the next value depends only on the current state.

For the estimation of the HMM parameters, the approach proposed in [141] will be used where it was shown that the approach archives high accuracy. According to this approach, the binary trace of an e2e path is first divided into partitions according to two patterns. These patterns are called the *good* and *bad* pattern. The *good* pattern starts with 0 and contains none occurrences of 1s. The *bad* pattern starts with a 1 and ends by a 0s burst of length equal to the pattern window (PW). When two 1s are separated by a 0s burst of length shorter than PW, then these 1s are considered to be part of the same *bad* pattern. The size of PW is choosed to equal the mean plus standard deviation of event bursts in the original trace as suggested in [134]. The second step in estimating the HMM parameters is to form two subtraces from the *good* and *bad* partitions called the *good* and *bad* subtraces. Third, the *good* subtrace is modeled using a state that produces only 0s while the *bad* subtrace is modeled using a 3rd-order general Markov model. Fourth, a state trace is created by replacing *good* partitions in the original trace by 0s and *bad* partitions by 1s. Lastly, the state trace is modeled using the Gilbert model.

7.2 Accuracy of Path Models

The previously mentioned, models can capture different degrees of temporal dependency between packets and the HMM can model paths with different events' patterns [140], [141]. More specifically, the Gilbert model, the EGM, and the 3rd order General Markov model (3rd-order GMM) are capable of capturing only the temporal dependencies between the states of successive packets. Unlike the latter two models, the Gilbert model is capable of capturing only the dependency between two consecutive packets. On the other hand, the HMM model can capture both, the temporal dependencies as well as the different event patterns of an Internet path. Therefore, it is necessary to determine the accuracy of the model in capturing the statistics of a given path. In addition, this also allows comparing the computational complexity (which depends on the number of the model states) with the achieved accuracy between the different models.

Similar to [141], the CDF of event-free and event bursts will be used as a measure to investigate models accuracy in capturing path statistics. More specifically, an artificial trace from a given model is first generated. Then, the *cc* between the CDFs of event-free and that between event bursts of the artificial trace and the original trace are calculated (the *cc* between event-free bursts' CDFs and the *cc* between event bursts' CDFs). The closer the *cc* value to 1, the higher the accuracy of the model in representing the statistics of the event and event-free bursts of the original trace. According to [134], *cc* values of 0.96 or lower indicate low accuracy of the model in capturing the statistics of the path being investigated.

Table 7.1 Correlation Coefficient, *cc*, of event and event-free bursts' CDFs of artificial traces (generated by the different models) and original traces.

VNs	Path	GM		EGM		3rd-order GMM		HMM	
		cc_1	cc_0	cc_1	cc_0	cc_1	cc_0	cc_1	cc_0
Lemgo and *Frankfurt*	*(1,2)*	0.9126	0.9911	0.9909	0.9829	0.9948	0.9918	0.9878	0.8553
	(2,1)	0.9988	0.9876	0.9816	0.9924	0.9999	0.9938	0. 9995	0. 9840
	(2,2)	1	0.9984	1	0.9982	1	0.9982	0.9999	0. 9989
Lemgo and *Milan*	*(24,16)*	0.9999	0.9982	0.9999	0.9987	0.9999	0.9991	0. 9999	0. 9989
	(24,17)	1	0.9990	1	0.9990	1	0.9991	1	0. 9991
	(25,16)	0.9986	0.9661	0.9995	0.9659	0.9995	0.9862	0. 9994	0. 9928
	(25,17)	0.9985	0.9725	0.9995	0.9723	0.9992	0.9877	0. 9993	0. 9930
	(26,16)	1	0.9995	1	0.9995	1	0.9994	1	0. 9995
	(26,17)	1	0.9992	1	0.9991	1	0.9992	1	0. 9991

To illustrate the approach, the accuracy for the previously mentioned models in representing the statistics of the e2e paths between *Lemgo* and *Frankfurt* VNs in the evaluation s*etup 1* (Section 6.4.1) and e2e paths between *Lemgo* and *Milan* VNs in the evaluation s*etup 2* (Section 6.4.1) is evaluated. The selection of these two sets is due to their different characteristics regarding event statistics and length of event bursts. By revisiting Figure 6.2 and Tables 6.7 and A.4, these sets of paths are provided in Table 7.1 with the results of the accuracy test. In the table, cc_1 is the *cc* between the CDFs of event bursts of the artificial trace and the original trace. By contrast, cc_0 is the *cc* between the CDFs of event-free bursts.

As given in Table 7.1, the different models have different accuracy in capturing the statistics of the different paths. Where, for each path, the accuracy is obtained from the mean of cc_1 and cc_0. The 3rd-order GMM is the only model that accurately models all paths with cc_1 and cc_0 greater than 0.99. Nevertheless, the Gilbert model has also high accuracy in modeling almost all paths (with a mean for cc_1 and cc_0 greater than 0.96). The only case where the Gilbert model does not yield high accuracy is for the path (*1,2*). This is attributed to the bursty nature of the path events. This observation motivates the use of the Gilbert model due to its low number of states (low computation complexity) when the events do not have a bursty nature.

7.3 Impact of Frequency and Type of Probing Packets

It was indicated in previous studies [11] that using different types of packets will yield different estimations for e2e paths unavailability. More specifically, TCP-based test packets are expected to provide more accurate estimations than ICMP-based test packets. This is mainly because ICMP packets might be dropped by some firewalls and some routers might treat ICMP packets with lower priority. In addition, the test packets represent discrete probes that are used to estimate the unavailability as well as the parameters of the MC models of the different e2e paths. A low probing rate might not capture short unavailability periods and reduce the accuracy of the path model. By contrast, a high probing rate is expected to estimate unavailability periods and parameters of model more accurately, but at the cost of high overhead. Therefore, the

impacts of the used probing packets' frequency and type in the measurements in this work are evaluated. For this evaluation, some sources and destinations from setups *1* and *2* in Section 6.4.1 were selected. For clearance of description, the selected source and destination nodes are depicted again in Figure 7.6.

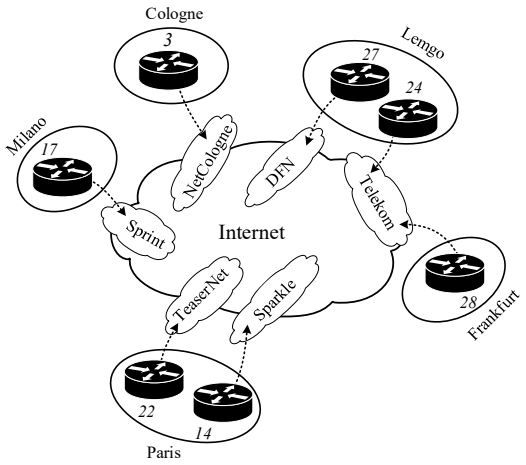


Figure 7.6 Location and access ISPs of the used nodes to analyze the impacts of test packets' frequency and type.

For each of the destination nodes in the figure, TCP probes were sent according to two frequencies, every 1 and 5 s, while ICMP ones were sent every 5 s. The measurements started on 2016-08-19, at 5 pm and ended on 2016-08-25, at 6 am. In the logging files of source *27*, some gaps (time intervals with no logged information) were found. These time intervals in the logging files of source *24* were not considered during the analysis.

Table 7.2 Unavailability of different e2e paths using different probing packets' frequency and type.

Path	**TCP**				**ICMP**	
	1 s		**5 s**		**5 s**	
	$u(\theta)$ **(%)**	$u(\theta)$ **(s)**	$u(\theta)$ **(%)**	$u(\theta)$ **(s)**	$u(\theta)$ **(%)**	$u(\theta)$ **(s)**
(24,28)	0.1022	445.2070	0.0023	9.8258	33.6050	146382.1231
(24,3)	0.0975	424.6908	0.0023	9.9662	0.1146	499.2218
(24,17)	0.1478	643.6772	0.0127	55.4208	0.1605	699.2021
(24,14)	0.5112	2226.9358	0.0182	79.0664	0.9759	4251.0868
(24,22)	0.1409	613.6951	0.0149	64.8738	0.1617	704.4983
(27,28)	1.1351	4944.5275	0.3683	1604.4472	37.4296	163041.9412
(27,3)	1.2013	5232.6327	0.3303	1438.7935	1.3542	5899.0098
(27,17)	1.1834	5154.7593	0.3661	1594.8415	1.4122	6151.4386
(27,14)	1.1866	5168.7385	0.2981	1298.4878	1.3170	5736.8872
(27,22)	1.2480	5436.3751	0.3200	1394.0723	1.7207	7495.1451

As shown in Table 7.2, ICMP probes overestimate the unavailability of most e2e paths by approximately one order of magnitude. At first examination of unavailability results estimated using TCP probes, 1 s probing frequency seems to estimate higher unavailability compared to the 5 s frequency. However, it was found that most unavailability periods are corresponding to isolated events (isolated occurrences of 1 s). One reason for such singular events might be packet errors that cause routers to drop it.

With this regard, such unavailability events might be packet based and a higher probing frequency might not detect other events (i.e. before or after the detected event).

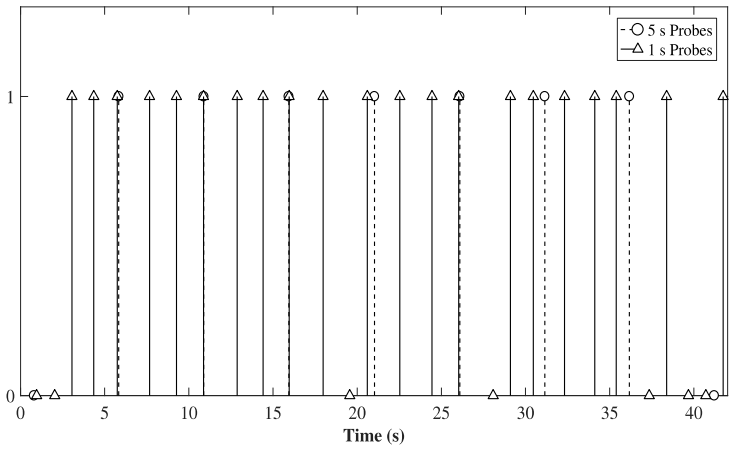


Figure 7.7 Unavailability as detected by the TCP *Ping* scripts of 1 s and 5 s.

In addition to isolated events, clustered events (not necessarily in sequence as described in Section 7.1.5) were also observed. Such clustered events might be caused by congestions or routing protocols failures that usually last for several seconds and impact subsequent packets [49]. When considering both log files for the 1 and 5 s probes, it was observed that such unavailability periods are present in both. More specifically, unavailability periods impacting several packets are likely to be detected using the 5 s probing frequency. This observation is also shown in Figure 7.7 where acknowledged and unacknowledged probes (1 s and 5 s probes on the path (*27*,*17*)) after the time point on 2016-08-24, at 08:46:39, are represented by stems. In the figure, the unacknowledged test probes are represented by stems at the value of one while acknowledged probes are represented by stems at the value of zero. Stems with triangles are for the probes of 1 s while those with circles are for the probes of 5 s. It is necessary to mention here that due to a coding bug, the probing frequency of the 1 s TCP *Ping* script during unavailability periods was not kept at 1 s (became about 2 s because of the selected timeout interval). Therefore, a fewer number of samples is observed for the probes in the figure than the expected one with 1 s frequency.

From the above-mentioned observations in this section, it is clear that the higher the probing frequency the higher the accuracy of e2e path unavailability approximation. However, this will result in a high monitoring overhead for the RC4CPS approach. On the other hand, a low probing frequency might not detect short unavailability events. Therefore, it is proposed here to use a probing frequency that is relative to the frequency of actual data transmission. This is also motivated by the fact that many industrial networks are characterized by having small packets and periodic traffic [104].

7.4 Comparison Test

In Chapter 5, the diversity estimation using correlation for RC4CPS was introduced. In this section, the possibility to identify pairs of e2e paths that do not share networks using the comparison test is examined. More specifically, the absolute M_x values for those paths that traverse different networks will be compared to those that share one or more networks. This because $|M_x|$ values reflect the degree of correlation as indicated in [70]. Moreover, the $|M_x|$ values will be compared with the results obtained using the *Traceroute* tool. As mentioned previously in Chapter 6, the diversity evaluation using the *Traceroute* tool is done at the network level. Consequently, path pairs that share one or more networks might also have very low values for M_x as those paths might traverse different links. Even in the case of shared links, the authors in [122] indicated that shared links were detected correctly in about 80% of the considered cases using the comparison test. In the remaining cases, the authors attribute the inability of the comparison test to detect shared links to the very low utilization / high bandwidth of the links in the considered time intervals. This resulted in insignificant correlation of delays over the considered paths indicated by the very low absolute value of M_x.

7.4.1 Comparison Test and Traceroute Measurements

For this evaluation, the diversity results to *Milan* obtained in *Setup 2* of unavailability evaluation (Section 6.4) are used. The results for path pairs between *Lemgo* VN and the remaining destination VNs are provided in Appendix B. In Tables 7.3 and 7.4, the assigned numbers in Figure 6.2b are used to identify the different nodes. First, the *Traceroute* results listed in Table 7.3 are considered. As it can be seen, the paths (*24*,*16*) and (*25*,*17*), for example, between *Lemgo* and *Milan* VNs are likely to be disjoint. In contrast, the paths (*24*,*16*) and (*24*,*17*) are joint and share the first a few hops in the *Telekom* network. Next, the comparison test is carried out between path pairs between *Lemgo* and *Milan* using the RTT delays measured during the same unavailability evaluation (Section 6.4). As listed in Table 7.4, path pairs that traverse different networks have very low absolute value for M_x. For the paths (*24*,*16*) and (*25*,*17*) as an example, $|M_x|$= 0.0044. On the other hand, the value of $|M_x|$ for the paths (*24*,*16*) and (*24*,*17*) that share a source interface equals 0.2484. As mentioned previously in Section 7.4, $|M_x|$ is not always significant for path pairs that share one or more networks as in the case of the paths (*26*,*16*) and (*26*,*17*). This agrees with the reasoning provided in Section 7.4 as the destinations used in the evaluation are ISPs routers that are expected to be connected to links with high bandwidth. The results presented in this section and in Appendix B can be summarized as follows. A total of 96 pairs of e2e paths were considered. The number of pairs that do not share a source or a destination interface (no shared networks) is 42. In contrast, the number of pairs that share a source or a destination interface (share one or more networks) is 54. The $|M_x|$ values for the 42 pairs with no shared networks ranged between 0.0001 and 0.0108. On the other hand, $|M_x|$

values for the 54 pairs with one or more shared networks ranged between 0.0005 and 0.9542. In addition, about 30% of the 54 path pairs with shared networks have comparable values of $|M_x|$ to those of the 42 path pairs with no shared networks. This shows that pairs of paths that traverse different networks can be identified using the comparison test with a good percentage of success. To avoid selecting subsets of e2e paths with only two paths that share one source or one destination interface, RC4CS use the conditions given in (5.14).

Table 7.3 Diversity results between *Lemgo* and *Milan* VNs (*Setup 2* in Section 6.4) using the *Traceroute* tool.

Source Node	Destination Node	ASs	Networks
24	16	AS3320, AS6453	Telekom, TATA
	17	AS3320, AS1239	Telekom, Sprint
25	16	AS3209, AS1273, AS6453	Vodafone, CW, TATA
	17	AS3209, AS1273, AS1239	Vodafone, CW, Sprint
26	16	AS6830*, AS6453	Unitymedia, UPC, TATA
	17	AS6830*, AS51531, AS1239	Unitymedia, UPC, DE-CIX, Sprint
* Unitymedia and UPC Austria are subsidiaries of Liberty Global telecommunications company and their checked IPs belong to the same ASN			

Table 7.4 M_x values for the 2-path subsets between *Lemgo* and *Milan* VNs (*Setup 2* in Section 6.4).

Subset	M_x	Subset	M_x
{(24,16),(24,17)}	0.2484	*{(24,17),(26,17)}*	0.0045
{(24,16),(25,16)}	0.1914	*{(25,16),(25,17)}*	0.4931
{(24,16),(25,17)}	-0.0044	*{(25,16),(26,16)}*	0.1865
{(24,16),(26,16)}	0.1827	*{(25,16),(26,17)}*	-0.0033
{(24,16),(26,17)}	-0.0022	*{(25,17),(26,16)}*	0.0017
{(24,17),(25,16)}	-0.0051	*{(25,17),(26,17)}*	0.0049
{(24,17),(25,17)}	-0.0039	*{(26,16),(26,17)}*	0.0109
{(24,17),(26,16)}	-0.0024		

8 Implementation Considerations

Although that various MP communication protocols were proposed, reliable communication in the Internet requires a specific set of features. In this chapter, the assumptions made before utilizing the RC4CPS approach are first described. Then, a number of requirements for reliable MP communication using RC4CPS are defined. After that, the fulfillment of the defined requirements by the different MP protocols considered in Section 3.4 will be evaluated. The main objective of this evaluation is to find the most suitable candidate from the existing MP protocols to implement RC4CPS. As mentioned previously, RC4CPS is not protocol dependent. However, implementing it using existing protocols will reduce the implementation complexity and avoid reinventing the wheel. Several of these protocols are already standardized. This will contribute to finding new domains for existing standards and motivate wide adoption for them.

8.1 Assumptions for Utilizing RC4CPS

As mentioned previously, the Internet has an uncontrolled and non-transparent infrastructure with constantly evolving/changing infrastructure/routing rules. Therefore, the RC4CPS approach was proposed in such a way that it considers the Internet as a black box and assists its e2e paths only from the end points. However, it was necessary to make a few assumptions with this regard.

The main assumption made for utilizing the RC4CPS approach is the availability of multiple ISPs for the source and for the destination. With such assumption, the sender and destination are expected to have multiple e2e paths that have different access ISPs. A related assumption with this regard is the availability of multiple and diverse paths in the core parts of the Internet between the different access ISPs. This inherited assumption is motivated by the results provided in [132]. The measurement results presented in Chapter 6 of this dissertation indicate that such assumptions are valid.

Another important assumption regarding the RC4CPS approach has been first introduced in Chapter 5. More specifically, it was assumed that there is at least one subset of e2e paths between the communicating parties that provide the required availability. Based on the measurement results provided in Chapters 6, 10, 11, this assumption is also valid.

It is also assumed that the communicating nodes are 100% reliable with no faults impacting them internally or externally. This assumption was made because the focus of this research is on the communication network, which is the Internet in this case.

8.2 Requirements for implementing RC4CPS

In this section, six requirements for RC4CPS are described. Not all of them are essential for the implementation for RC4CPS. However, when all requirements are satisfied, the reliability gains and the feasibility for wide deployment in the future are maximized.

8.2.1 Active Multihoming

Redundancy is one of the means that is widely utilized to improve reliability. This is also utilized in RC4CPS along with dynamic online selection of multiple e2e paths to improve reliability. To achieve this, RC4CPS requires the use of multiple network interfaces with access to different ISPs. Hence, support for multihoming is required with the capability to use all interfaces simultaneously (active multihoming).

8.2.2 Approach Layer in the OSI Model

Reliable MP communication in the Internet needs to be realized at the transport layer or above. e2e solutions are usually implemented at these layers to provide the different functions and services over the Internet. Approaches at these layers still have pros and cons when compared to each other. Transport layer protocols can provide end-systems with path statistics about time delay, packet loss, throughput, etc. As a result, they can use their access to this fine-grained information about the network's e2e characteristics to enhance the utilization of multiple paths for load balancing, bandwidth aggregation, and fault tolerance. In addition, transport layer solutions can provide further functionalities such as CC, flow control, and packet reordering. Nevertheless, transport protocols might require a modified kernel (especially for new protocols) and carful design to avoid any incompatibility issues with middleboxes. Therefore, changes to transport protocols also involve changes to the host OSs and thus impede quick deployment.

Application and session layers' solutions are located above the OSs kernel, hence, they do not require modifications to OSs. Solutions at these layers simplify the addition or removal of user-driven functionalities including MP communication mechanism such as MP selection, packet duplication, packet reordering, etc., but at the cost of additional overhead and computing resources. This is mainly because they have no information about the underlying network topology or the path statistics, and thus rely on custom solutions. This in turn makes the realization of reliability functions and CC is unfavorable. In addition, application layer designs are generally implemented for specific applications and therefore lack universal deployability. By contrast, session layer approaches make use of specialized middleware or virtual sockets to provide MP communication APIs between the application and transport layers.

8.2.3 Data Duplication

Packet duplication over the selected e2e paths is another requirement to maximize reliability when using RC4CPS. Here, packet copies from other paths instantly resolve unavailability events and compensate for lost packets on other paths.

8.2.4 Path Selection

The selection of the minimum set of e2e paths in RC4CPS to provide a certain level of communication reliability is dynamic. This is required because the reliability of an Internet e2e path might change over time due to, for example, congestions. In addition, path selection enables end-systems of selecting paths that, for example, traverse different networks. Therefore, it is important to have control over the path selection process during and after connection establishment, without having to tear down and restart the communication session.

8.2.5 Compatibility with Middleboxes

Deploying a communication protocol in the Internet requires compatibility with the various types of middleboxes. This is because some middleboxes, such as firewalls, work based on a white list approach. Only allowed flows where the packets have known contexts and structures are able to pass. The development of stateful firewalls increased the complexity of deploying new protocols. They consider a packets context by validating sequence numbers, ports, and IP addresses and match them with known and active connections. This also necessitates using two sequence number spaces for TCP-like approaches [71]. One for each individual subflow and the other for the overall data transmitted. The data sequence number is used mainly for data reordering at the receiver. By contrast, subflow sequence numbers are used to ensure compatibility with middleboxes and, therefore, need to be continuous. Otherwise, subflow packets might cause middleboxes to discard them or request unnecessary retransmissions.

8.2.6 Fairness

The Internet is a public network with shared infrastructure and best-effort type of service. The use of multiple subflows per connection might lead to an unfair allocation of resource in bottleneck links. However, and regardless of the utilized protocol, each Internet flow need to be TCP-friendly as TCP is the dominant transport protocol in the Internet [71]. This can be considered as a standard in the Internet to provides fairness for the different flows. Even though CPSs will have industrial networks where traffic is characterized by having small packets of periodic nature [104], the MP approach should consider fairness for wide scale deployment in the future.

8.2.7 Open-source Development

Open-source protocols allow researchers to use and adapt existing protocol functionalities or implement and test additional ones. Open-source solutions are usually developed more rapidly, with a large number of contributors around the world.

However, this is not an essential requirement for reliable MP communication. Nevertheless, it would be a helpful reference for future research in this domain.

8.3 Evaluation of MP Protocols

Table 8.1 presents a summarized overview of which requirements are met by the considered MP approaches in Section 3.4. Further details are elaborated in the following sub-sections which are organized based on the OSI layers of the approaches.

8.3.1 Transport layer

Active Multihoming:

From the listed approaches, SCTP and DAR-SCTP are the only protocols that use a single e2e path at a time to transmit data. SCTP is multihomed in the sense that it is capable of supporting multiple network interfaces and making handover of connections between paths. This is done also without reestablishing the association. By contrast, all other approaches aim at increasing throughput or reliability by establishing multiple concurrently communicating paths. Nevertheless, most of them split the data over available paths, with no capability of switching paths or replicating packets to improve reliability.

Data duplication:

Most of the considered propositions for MP communication target improving data throughput. Therefore, the attained increase in reliability is usually just a side effect due to the decentralized points of failure. NC-MPTCP, SC-MPTCP, FMTCP, and MPLOT protocols focus on aggregating bandwidth but provide a certain degree of redundancy by using FEC coding to recover from intermittent packet losses. However, in case of heavy congestion, these protocols still perform retransmissions with the extra cost of packet encoding overhead. In addition, they also require support of FEC coding by both communicating parties. M/TCP and R-M/TCP perform packet duplication only in case of packet losses to minimize retransmissions. The only approaches encountered in the considered work that focus solely on maximizing redundancy through duplicating all data over multiple paths are the iPRP and M-TCP. In addition, iPRP offers a dedicated control plane that manages the data transfer between transport and application layers to discard duplicated packets at the receiver. M-TCP, on the other hand, processes every packet duplicate which results in excessive delays and further performance degradations.

Path selection:

As mentioned previously, it is needed to have dynamic path selection due to the fluctuating path reliability in the Internet. MPTCP uses a path manager that adds and removes new or inactive interfaces from a connection. However, the protocol provides

only a full mesh configuration of paths between available interfaces and there is no way for an end-user to control it (for example, to make the selection of paths based on

Table 8.1 Evaluation of MP protocols fulfillment of the requirements for implementing RC4CPS.

Protocol	Active multihoming	Data duplication	Path selection	Middleboxes	Fairness	Open-source development
Transport layer approaches						
SCTP	x	x	✓	x	✓	✓
CMT-SCTP	✓	x	✓	x	x	✓
CMT/RP-SCTP	✓	x	✓	x	✓	x
DAR-SCTP	x	x	✓	x	x	✓
FPS-SCTP	✓	x	✓	x	x	✓
MPTCP	✓	x	x	✓	✓	✓
Yang & Amer	✓	x	x	✓	✓	✓
NC-MPTCP	✓	x	x	✓	✓	x
SC-MPTCP	✓	x	x	✓	✓	x
FMTCP	✓	x	x	✓	✓	x
CWA-MPTCP	✓	x	x	✓	✓	x
MPTCP-SPA	✓	x	x	✓	✓	x
QoS-MPTCP	✓	x	x	✓	✓	x
OpenFlow-MPTCP	✓	x	✓	x	✓	x
A-MPTCP	✓	x	✓	x	✓	✓
MPCubic	✓	x	x	✓	✓	x
pTCP	✓	x	x	x	X	x
cTCP	✓	x	x	x	X	x
M-TCP	✓	✓	x	x	X	x
M/TCP	✓	x	x	x	X	x
R-M/TCP	✓	x	x	x	X	x
mTCP	✓	x	✓	x	✓	x
iPRP	✓	✓	✓	✓	X	✓
E2EMPT	✓	x	x	✓	x	x
R-MTP	✓	x	x	x	x	x
MPLOT	✓	x	x	x	x	x
Application layer approaches						
GridFTP	x	x	x	✓	x	✓
MultiTCP	x	x	x	✓	x	x
PSockets	x	x	x	✓	x	x
XFTP	x	x	x	✓	x	x
MPRTP	✓	x	x	✓	x	✓
MPTS-AR	✓	x	✓	x	x	x
Session layer approaches						
ATLB	✓	x	x	✓	x	x
PATTHEL	✓	x	x	✓	x	x
DBAS	✓	x	✓	✓	x	x
OPERETTA	✓	x	✓	✓	x	x
RI2N/UDP	✓	x	x	✓	x	x
MuniSocket	✓	x	x	✓	x	x

reliability). A-MPTCP, OpenFlow-MPTCP and mTCP use overlay networks to prevent that and allow users with special network controllers to manage selected paths. But that requires additional and modified network equipment between end-systems. This increases complexity and reduces future scalability of the approach. Regarding MPTCP, a socket API is currently being developed in [142], which is supposed to give users control over path management. A similar socket API extension was already proposed and implemented for SCTP [143]. By contrast, path selection based on connected networks to the different interfaces is allowed by default in iPRP through its Network subcloud Discriminators (INDs). More specifically, iPRP compares the INDs of the source and destination nodes and establishes paths only between matching INDs. However, such a path selection will need to be adapted to the Internet. The other approaches do not further consider path selection. They mostly act autonomously and form a full mesh between all available interfaces by default. cTCP is designed to consider upper layers selection setups, but no further details were provided on how this should be implemented.

Middleboxes:

Most of the presented approaches at the transport layer are based on the legacy TCP or UDP protocols. They attempt to keep their individual communication subflows as close as possible to regular TCP or UDP flows. Except MPTCP variants and pTCP, all other TCP-based designs do not utilize a second sequence number. As a result, the utilization of these protocols in today's Internet might not be feasible. SCTP is implemented in most popular OSs, but uses a different protocol identifier. Unfortunately, most firewalls do not support its protocol identifier. Similarly, the functionality of pTCP's depends on a modified header wrapped around the standard TCP header which is not compatible with most middleboxes. MPLOT uses an additional sequence number to mark the position of packets in the individual FEC blocks for correct erasure coding. M/TCP uses a Route Id to allow acknowledgments to be sent over different reverse paths and, consequently, to be identified correctly. However, both protocols do not utilize continuous sequence numbers for the data on their subflows. Besides the usual requirement of a modified kernel for the transport layer solutions, mTCP and A-MPTCP also require RON and LISP network support respectively. Similarly, OpenFlow-MPTCP requires OpenFlow-enabled routers and switches. This in turn necessitates that the middleboxes be aware of such additional protocols and their traffic such that these protocols work properly.

Fairness:

To provide fairness, MPTCP initially controlled overall aggressiveness of its MP subflows by making use of the Linked Increases Algorithm (LIA) in the Coupled Congestion Control (CCC) scheme [144]. To further improve the fairness of MPTCP under a wider spectrum of circumstances, Opportunistic Linked Increases Algorithm (OLIA) [145] was proposed. Regular SCTP uses slightly modified TCP CC that

provides fairness towards regular TCP flows. This is not the case for CMT-SCTP or other variants that use paths concurrently. CMT/RP-SCTP was proposed to address this problem and implements a CC scheme that is aware of the paths' interaction. With exception of mTCP, all other TCP based approaches use a separate CC for each path individually, reducing their fairness towards other TCP flows. This is done to control the transmission rates of each path separately such that, for example, a Forward Prediction Scheduler can send data out of order such that they arrive at the receiver in the right order. To reduce this unfairness, different solutions were adapted. mTCP uses a shared congestion detection mechanism to suppresses paths with shared congestion. MPLOT uses Explicit Congestion Notification (ECN) in conjunction with ECN-enabled middleboxes in order to detect impending congestion and lower the transmission rate. iPRP uses UDP as its protocol substrate that naturally does not offer any CC and is not aware of other communication flows.

Open-source development:

As previously mentioned, the most established protocols for multihoming are the SCTP and MPTCP ones. Therefore, kernel implementations for these two protocols are available in Linux and FreeBSD. In contrast, only a few of their extensions are also implemented. In addition, most of their extension as well as other designs are simulated in the network simulator *ns-2* [146]. But these *ns-2* implementations are not published and, therefore, hinder any future development or thorough evaluation of the features implemented. iPRP is available as a user-space Linux kernel implementation, but is still under development [147].

8.3.2 Application Layer and Session Layer

The application and session layers have similar attributes and features and, therefore, they are presented in this section together. MP solutions at these layers do not modify the transport protocol stack. In order to have information about the network topology or the path statistics, these approaches rely on custom solutions. As a result, the communication overhead for such approaches is higher when compared to those approaches at the transport layer. On the other hand, they are located above the OS's kernels and, therefore, simplify the addition or removal of user-driven functionalities like path selection.

Active multihoming:

Except GridFTP, MultiTCP, PSockets, and XFTP, all presented application and session layer approaches support active multihoming and use their paths concurrently.

Data duplication:

Only MPTS-AR and MPRTP at the application layer consider data duplication. MPTS-AR offers to choose between concurrent or redundant transmission modes. This is done by setting the corresponding bits in a field called the Path Usage Method (PUM). The

field is part of the exchanged OpenPath messages used for path allocation. In the MPRTP specifications, it is stated that alternate paths should also be used for sending retransmitted or redundant packets. However, neither MPRTP nor MPTS-AR have these features implemented yet. The realization of the redundancy model itself is also not specified. On the session layer all designs focus on aggregating bandwidth and do not offer data duplication.

Path selection:

MPRTP and MPTS-AR use their subflow reports to adjust their path configuration dynamically. MPRTP closes or creates new paths, while MPTS-AR just uses its relay controllers to reroute subflows over different relay servers. With this regards, MPTS-AR utilizes the OpenPath protocol for relay paths configuration as its API. By contrast, It is not indicated whether MPRTP protocol has an API to control path selection. The other presented application layer approaches establish multiple flows over single e2e path and, therefore, do not support e2e MP selection. At the session layer, the assignment of applications to certain network interfaces to provide a simple path selection interface is enabled in DBAS and OPERETTA. The rest of the session layer approaches utilize by default a full mesh configuration between all interfaces.

Middleboxes:

As the application and session layer approaches do not modify the lower layers of the protocol stack, they do not interfere with middleboxes. MPRTP, MPTSAR, RI2N/UDP and MuniSocket use UDP as their underlying transport protocol while the rest use TCP. MPTS-AR uses an overlay network with a complex mesh of relay servers and controllers to manage MP routing. This also requires middleboxes that are aware of the additional control protocols and their traffic nature.

Fairness:

GridFTP, MultiTCP, PSockets, and XFTP use only a single e2e path to create multiple TCP flows and provide higher throughput. While most MP protocols make use of disjoint paths to increase bandwidth, these approaches rather circumvent TCP CC to gain more bandwidth. This contradicts with the concept of TCP-fairness and is expected to degrade other users throughput. As MPRTP, RI2N/UDP, MPTS-AR, and MuniSocket utilize the UDP protocol, they do not provide any CC. However, MPRTPs control protocol RTCP, which runs in parallel to RTP subflows, can obtain path statistics such as jitter and RTT. The information is used to detect shared congestion between subflows and to determine the load distribution across all paths. Subflow Sender Reports (SSR), which work similar to MPRTP's RTCP, were suggested for MPTS-AR to handle congestion on its paths. They inform the user agent in the sender about its subflows delivery quality to enable path reconfiguration and their corresponding load distribution. This enables MPRTP and MPTS-AR to provide more fairness towards regular TCP flows. By contrast, session layer approaches based on

TCP do not consider the coupling of CC on their subflows. Consequently, they do not provide fairness to regular TCP-based flows.

Open-source development:

Only GridFTP and MPRTP are open-source and available through the Globus Toolkit [148] and as a GStreamer-implementation [149] respectively. It is worth mentioning that MPRTP and MPTS-AR are in an early development stage and were partially simulated to evaluate them. However, many functional goals are still not realized.

8.3.3 Selected MP protocol Candidate for RC4CPS

After evaluating this wide spectrum of MP protocols with regard to RC4CPS requirements, it is clear that iPRP represents the most fitting candidate. Unlike other protocols, it requires the least amount of adaptation and supports all requirements except TCP-fairness. This issue can be elevated if the selected paths are disjoint. Nevertheless, a number of issues still need to be addressed in order to realize RC4CPS using iPRP. For example, iPRP was proposed for dedicated IP networks. Therefore, its utilization in the Internet might require further adaptation.

9 Confidence Interval for MP communication Unavailability

In Chapter 6, I evaluated the unavailability of 2- and 3-path subsets. According to the estimated values, the unavailability was in the range 0-0.044%. If the scope is limit to 2-path subsets and unavailability estimation using TCP-based *Pings*, then the range is 0-0.0012%. From such evaluation, it seems that MP communication will support the high communication availability needed for Internet-based CPSs such as smart grids. However, such estimation is based on a finite sample data that come from a finite set of e2e path. Hence, an important question with this regard is: How good is the estimation of the unavailability of MP communication when considering the population of all e2e path pairs in the Internet. In other words, if the mean unavailability for all 2-path subset samples considered in *Setup 2* in Section 6.4 is $\hat{\mu}$, then how much close $\hat{\mu}$ to the population mean μ? Due to sampling variability, $\hat{\mu}$ and μ might never be the same. A similar question is raised regarding subsets of paths with more than two e2e paths.

Fortunately, this issue can be addressed using CIs [150], [151]. Such intervals provide a range of values where the parameter of interest is likely to be contained. As the estimation is based on only a sample set from the full population, it is not certain that the estimated interval will contain the true population parameter. Nevertheless, the construction of the confidence interval is done in such a way that a high confidence is asserted about its inclusion of the investigated parameter.

In this chapter, the methods to obtain the CI from the sample data and their assumptions are first described. After that, the sample data for estimating the true mean of MP unavailability is presented. The fulfillment of required assumptions by the sample data set is also discussed. Lastly, the estimated CI regarding the unavailability of MP communication using 2-path subsets is presented.

9.1 CI Estimation Methods

The approaches for estimating CIs are categorized into parametric and non-parametric approaches. Parametric approaches are those based on a certain family of distributions described using specific sets of parameters. More specifically, they are usually assuming a certain distribution about the underlying population, namely the normal distribution. By contrast, the non-parametric approaches are distribution free and do not assume a certain distribution about the population except that it is continuous. Even this assumption is not necessary for CI estimation [151]. An Important assumption for parametric and non-parametric approaches is that the data samples are independent and identically distributed (i.i.d.) [150], [152]. This indicates that the sample data should be uncorrelated over time.

9.1.1 CI Using Parametric Approach

The procedure described in this section is usually referred to as the estimation of CI of a normal distribution with unknown variance [151]. From the name, it is clear that the procedure assume that the underlying population has a normal distribution. In practice, many populations are well described by a normal distribution. Therefore, such procedure has wide applicability. In addition, the procedure can still be used even when the sample data have slight to moderate departure from the normal distribution. However, if there is not enough evident about the normality of the population distribution, then a large number of samples is needed (e.g. 30 or more). In this case, it is possible to estimate the CI without the normality assumptions. This is based on the Central Limit Theorem that says: for an i.i.d. sample data of size n, the distribution sample means becomes closer to normality as n increases, irrespective of the underlying population distribution. If the sample data size is not large enough (less than 30), then the non-parametric approaches can be used.

Given a sample data of size n with X_1, X_2, …, X_n i.i.d. samples, sample mean $\hat{x}$, and variance s^2 correspondingly. Then, the $100\frac{\alpha}{2}\%$ CI on μ is:

$$\hat{x} - t_{\alpha/2,n-1}\, s / \sqrt{n} \le \mu \le \hat{x} + t_{\alpha/2,n-1}\, s / \sqrt{n}, \tag{9.1}$$

where (1-α) is the confidence coefficient and $t_{\alpha/2,\,n-1}$ is the upper percentage value (i.e. $100\frac{\alpha}{2}\%$) in the t-distribution table with n-1 degrees of freedom. The upper and lower confidence bounds on the mean can be also directly obtained from (9.1). The use of the t-distribution is attributed to the unknown mean μ and variance σ^2 of the normal distribution of the sample data. In this case, the random variable with the t-distribution of n-1 degrees of freedom is formed from the original sample data with sample mean $\hat{x}$ and variance s^2 such that:

$$T = \frac{\hat{x} - \mu}{s/\sqrt{n}}. \tag{9.2}$$

9.1.2 CI Using Non-parametric Approach

Non-parametric approaches include the Sign Test, the Wilcoxon Signed-rank Test, and the Wilcoxon Rank-sum Test [151]. In this section, only the Sign Test is described.

Sign Test:

The hypothesis considered by the Sign Test is about the median of a distribution. Given the distribution of a random variable X, then the median $\hat{\mu}$ is the value of X where $P(X \le \hat{\mu}) = P(X \ge \hat{\mu}) = 0.5$. The hypothesis to be tested might be for example:

$$\begin{aligned} H_0 &: \hat{\mu} = \hat{\mu}_0, \text{ and} \\ H_1 &: \hat{\mu} < \hat{\mu}_0, \end{aligned} \tag{9.3}$$

where H_0 is the null hypothesis that $\hat{\mu} = \hat{\mu}_0$ while H_1 is the alternative hypothesis. $\hat{\mu}_0$ is the expected median. The procedures of the test consist of first obtaining the differences:

$$X_i - \hat{\mu}_0, i = 1, 2, ..., n, \tag{9.4}$$

where X_i is the i^{th} random variable from the sample data. The H_0 hypothesis is true when the differences in (9.4) might be positive or negative but are equally likely. The test statistic as adopted in [151] is the number of positive differences, denoted by R^+. In this case, the Sign Test becomes a check on whether the value r^+ of R^+ is a value that comes from a random number with binomial distribution and parameter $p = 0.5$. Consequently the P-value, which is the probability of observing the current or more extreme results while H_0 is true, for the observed r^+ of the number of positive signs is calculated from the binomial distribution such that:

$$\text{P-value} = P(R^+ \leq r^+ \ when \ p = 0.5). \tag{9.5}$$

When the P-value is less than the significance level α, then H_0 is rejected and the alternative H_1 is true.

Another procedure to obtain the CI is to invert the Sign Test in the sense that it considers all possible null hypotheses. If all real numbers will be considered, then, an infinite number of null hypothesizes need to be tested. However, this is not needed because the sign of the difference in (9.4) will change only when $\hat{\mu}$ pass one of the data points [152]. Hence, H_0 is tested for each data point plus two exterior points to the widest interval represented by the sample data. As the investigated parameter is the median with $P(X \leq \hat{\mu}) = P(X \geq \hat{\mu}) = 0.5$, the Binomial distribution with parameters n and $p = 0.5$ is symmetric. As a result, the test needs to consider the possible intervals formed by the data points. For example, if the sorted sample data is X_1, X_2, X_3 , X_4, then two intervals from the data points can be formed, namely (X_1, X_4) and (X_2, X_3). The longest interval (X_1, X_4) fails to include $\hat{\mu}$ when $\hat{\mu}$ is not in the range of the sample data. In this case, all differences in (9.4) will have the same sign and r^+ can be either 0 or n. For the second longest interval (X_2, X_3), the probability that it does not include $\hat{\mu}$ is equivalent to r^+ of 1 or n-1. From this pattern, the confidence of each of the intervals formed from the sorted data takes the value:

$$1 - 2P(R^+ \leq k), \tag{9.6}$$

where k is the number of data points pairs outside the interval.

9.2 Sample Data for MP Unavailability

The derivation of the CI in this section considers only the 2-path subsets. This is mainly because the minimum number of paths to be used by RC4CPS is two. Moreover, almost all three path subsets had 0% unavailability in the different evaluations that were conducted. In general, the unavailability of MP communication is inversely proportional to the number of paths used. Therefore, the estimation of the CI of the mean of unavailability for MP communication is limited to the case of two paths.

For the derivation of the CI, the 2-path subsets from *Setup 1* and *Setup 2* in Section 6.4 and those from *Setup 2* in Section 10.2 are considered. The average unavailability values for the subsets from Section 6.4 are provided in Tables 6.8, 6.11, A.2, A.5, A.8, and A.11. The average unavailability values for the other subsets are presented in Section 10.2. As the measurements of e2e paths unavailability in Section 6.4 were carried out during the same time interval, such data sample might be correlated. Hence, direct utilization of theses samples is not recommended. To address this issue, it is necessary to obtain samples from the collected data but as a time series [150] and check the correlation between them using the lag-j correlation coefficient given as:

$$\rho_j = corr\,(x_i, x_{i+j}) \tag{9.7}$$

where x_{i+j} is the data sample with lag of j samples from x_i.

To obtain such time series, the unavailability of 12 hours intervals separated by 6 hours intervals is calculated. In each 12 hours interval, the average unavailability for each 2-path subset between *Lemgo* and one of the other VNs is calculated. Then, the sample with the highest average unavailability is considered. For example, let us consider the VNs *Lemgo* and *Cologne* as the starting pair in *Setup 2* in Section 6.4. Then, the unavailability time series is constructed as follows. First, $u(\theta)$ of each of the 2-path subsets satisfying (5.14) between these VNs over a time interval of 12 hours is calculated. Second, the highest value of $u(\theta)$ from the available subsets is considered. Third, an interval of 6 hours after the considered 12 hours interval is skipped. Fourth, a new pair of VNs is selected and the steps from one to three are re-done. Once all pairs are used, then the procedure is repeated again for the remaining time interval and starting from the first pair of VNS. Based on this procedure, a sample data size of 26 from Sections 6.4 and 10.2 is obtained. These samples plus four others that were obtained from additional measurements are listed in Table 9.1.

Now the correlation of the sample data from Section 6.4 presented in Table 9.1 is checked using (9.7). The lag-1 correlations for the samples from *Setup 1* and *Setup 2* are -0.12 and -0.07 respectively which indicate none or very week correlations. This can also be observed from the time lag plots of the samples shown in Figures 9.1 and 9.2. As the size of the sample data in Table 9.1 equals 30, then the normal distribution can be assumed and (9.1) can be used to calculate the CI. For the sample data in Table 9.1, the sample mean and standard deviations are 9.1×10^{-5} and 3.8×10^{-4} correspondingly. Therefore, the upper bound of the 95% CI on the mean of MP unavailability when using two paths is given as:

$$\mu \leq 2.3 \times 10^{-4}\%. \tag{9.8}$$

In other words, there is confidence that 95% of the mean values of MP unavailability when using two paths are $\leq 0.00023\%$.

To estimate the CI using the Sign Test, $\hat{\mu}_0$ is selected to be 10^{-5}. Then the number of positive differences, r^+, obtained using (9.4) is counted which is 13. In this case, the P-

value calculated as in (9.5) is 0.29. This value is larger than the significance level α = 0.05. Hence the null hypothesis that $\hat{\mu} = \hat{\mu}_0 = 10^{-5}$ is true and the alternative that $\hat{\mu} < \hat{\mu}_0$ is rejected.

Table 9.1 Unavailability samples.

Average Unavailability (*u(θ)*) for Samples		
Section 6.4		**Section 10.2**
Setup 1	***Setup 2***	***Setup 2***
0	0	0
0.00030	0	0.00005
0	0	0
0.00002	0.00001	0
0	0.00001	-
0	0.00003	-
0.00010	0	-
0	0	-
0.00208	0.00003	-
0	0.00001	-
0	-	-
0	-	-
0.00002	-	-
0	-	-
0.00002	-	-
0.00005	-	-

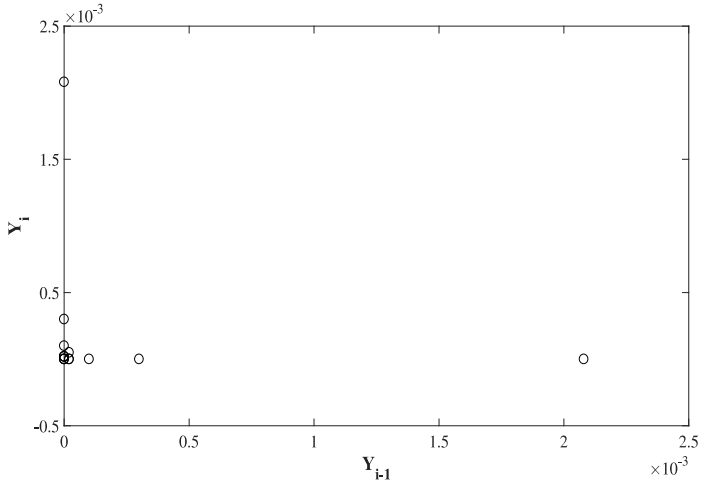


Figure 9.1 Lag-1 time-lag plot for samples from *Setup 1* in Section 6.4.

Lastly, the CI is estimated using the second procedure in Section 9.1.2. Based on that, four non repeated intervals are obtained using the samples in Table 9.1. These are presented in Table 9.2. If the CI with 93% considered from Table 9.2, then the upper bound of the CI is 3×10^{-4}.

By considering the wider CI from those obtained using the parametric and non-parametric approaches, MP communication using two paths is expected to provide unavailability in the range of 0–0.0003% (i.e. 99.9997-100% availability) with at least 93% confidence level. In addition, there is a confidence of 95% that the average

communication unavailability/availability using two paths is 0.00001%/99.99995%. Such availability levels support a wide range of smart grid applications [8]. Not to mention that in this derivation of an upper bound of the mean of MP unavailability, only subsets with two paths and with the highest average unavailability observed were considered. Hence, the upper limit on the mean value of MP unavailability for subsets with more than two paths is expected to be even lower than 3×10^{-4}. In fact, almost all three path subsets considered in the conducted measurements achieved 0% unavailability.

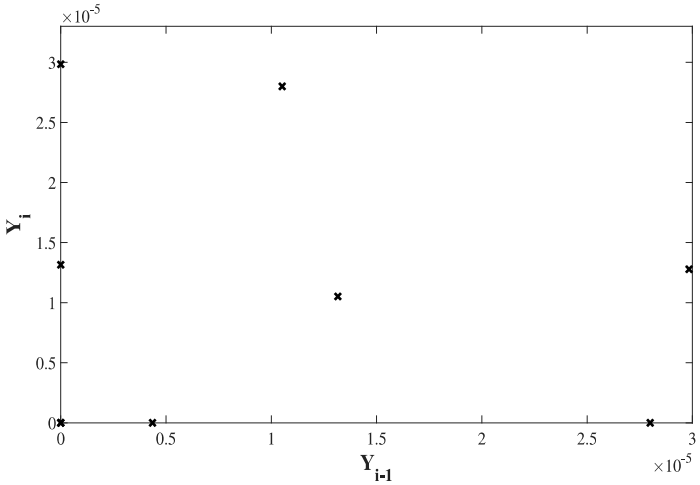


Figure 9.2 Lag-1 time-lag plot for samples from *Setup 2* in Section 6.4.

As a result, it is concluded in this chapter that MP communication unavailability using two e2e paths will support the unavailability requirements of 0.0003% or less with a confidence level of 93%. It is also expected that MP communication unavailability using more than two e2e paths will provide lower unavailability with a higher confidence level.

Table 9.2 Sample intervals and corresponding confidence levels.

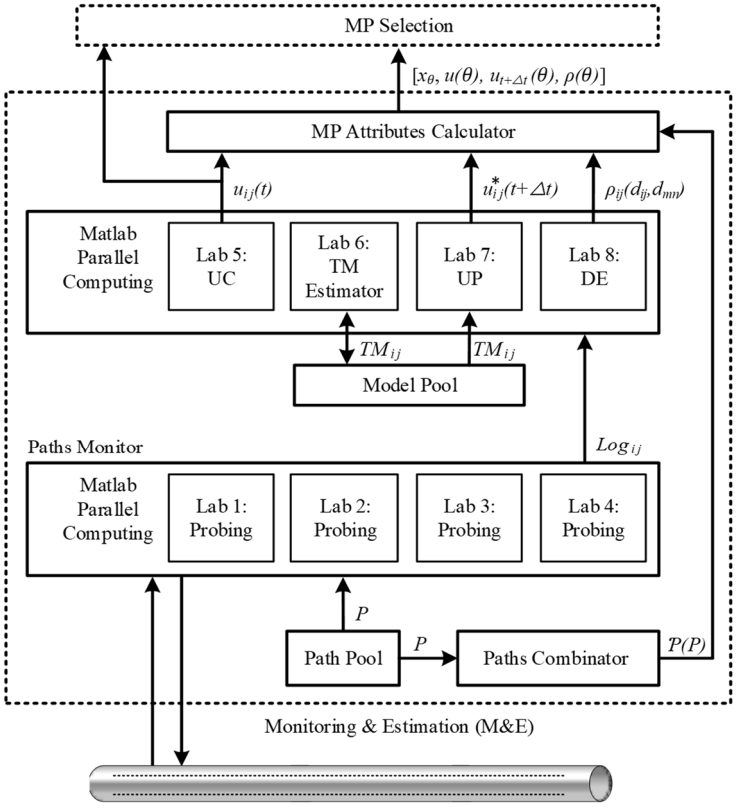
Sample Interval		**Confidence Level (%)**
0	0.00208	99
0.00001	0.00030	93
0.00002	0.00010	71
0.00003	0.00005	27

10 Implementation and Evaluation of RC4CPS Using MATLAB

In this chapter, an implementation of RC4CPS using MATLAB is described and evaluated. In the implementation, only the M&E and MP Selection components of the RC4CPS sender in Figure 5.3 are considered. The main objective here is to evaluate the online calculation of the e2e paths' metrics and the online decision making procedures of RC4CPS for choosing θ_{pr} and θ_{ba}. The implementation offers a fast deployment and test platform to characterize the online performance of RC4CPS before developing its transport layer implementation which will be described in Chapter 11.

10.1 Block Diagram of the Implementation

The block diagram of the implementation is shown in Figure 10.1. As illustrated, the Parallel Computing Toolbox™ of MATLAB is utilized to perform the different monitoring, estimation, and prediction functions at the sender part of RC4CPS. The use



Key:
Lab *i*: is a block of code that runs on a specific MATLAB worker (runs in parallel with other workers).

Figure 10.1 Implementation of RC4CPS using MATLAB.

of the Toolbox is to reduce the execution time of the code and to perform the MP selection during runtime. As illustrated in Figure 10.1, the set of e2e paths, *P,* is first provided to the Paths Monitor component where *P* Labs (MATLAB workers) probe the paths and update their logs. After that, the logs (Log_{ij}) and the selected model for each path are provided to the UC, UP, & DE components. Similarly, four Labs use the logs of monitored paths to (1) approximate $u_{ij}(t)$; (2) estimate the diversity of the different 2-path subsets using (5.5); (3) determine $u^{*}_{ij}(t+\Delta t)$; and (4) update TMs of the MC models of the monitored paths. Lastly, the AM and $u_{ij}(t)$ are provided to the MP Selection component to select θ_{pr} and θ_{ba}.

10.2 Evaluation

For the same reasons indicated in Section 6.1, the NorNet testbeds were not used for the conducted evaluations in this chapter. As the PlanetLab testbeds are single-homed, their utilization to run the MATLAB implementation of RC4CPS was not considered. Nevertheless, the utilization of these testbeds to form VNs as described in Chapter 6 was considered. Only in one case, which is described in Section 10.2.4, it was possible to find two PlanetLab testbeds in close proximity from one another. In many other cases, the issues described in Section 6.1 prevented forming VNs using the PlanetLab testbeds in other cities. Therefore, a personal computer (PC) with the MATLAB implementation of RC4CPS was utilized as a sender in this evaluation. By contrast, the destinations were VNs that consists of two or more end-systems in close geographical proximity from one another.

The start and end of unavailability events in this implementation are determined in a similar way to that presented in Section 6.4, but with a fixed probing frequency (every 5 s). The reason for adopting such stringent measure for the duration of unavailability events in this implementation is provided in Section 6.2.2. However, this was not adopted in the RC4CPS implementation using the iPRP protocol that will be presented in Chapter 11. More specifically, an unavailability event starts from the receive time of the ACK to the last successful probe and continues till the send time of the next successful probe.

In the following sections, two evaluation setups for RC4CPS are presented. The first one represents a first evaluation of the online decision making procedures of RC4CPS in real-world using a single-homed PC. In the second setup, the implementation was father tuned to allow the use of a multihomed PC connected to different access ISPs, namely DFN and Deutsche Telekom networks. This allowed including further scenarios and locations in *Europe* in the evaluation of RC4CPS performance. In both setups, the e2e paths are probed every 5 s and the maximum allowed unavailability, u_r, that should not be exceeded when using RC4CPS is set to 10^{-5}%. It is important here to describe the approach used for plotting the evaluation results in this chapter. As the probing is done every 5 s, plotting all data samples for all subsets of paths might result in unclear figures

especially in the case of $u_{t+\Delta t}(\theta)$ values. Therefore, markers were used and plotted in intervals such that the markers are not plotted on top of each other.

10.2.1 Evaluation Setup 1

In this initial evaluation, RC4CPS approach is evaluated using the setup shown in Figure 10.2. As illustrated, a PC (sender) with the MATLAB implementation of RC4CPS is connected through a PacketStorm network emulator [153] to the Internet. Four destinations in *Frankfurt*, *Germany*, that represent one VN with four interfaces connecting to different ISPs were considered. The access ISPs and the given interfaces' numbers are shown on the figure. Moreover, Table 6.1 describes the acronyms of the ISPs illustrated. Hence, this evaluation considers four paths each represented by a pair (*i*,*j*) (e.g. the path to the Sprint's destination is (*1*,*1*)). As the source has only one interface, the last two conditions in (5.14) were not considered in this evaluation. The PacketStorm is used in this setup to change the characteristics of the paths to observe the behavior of RC4CPS.

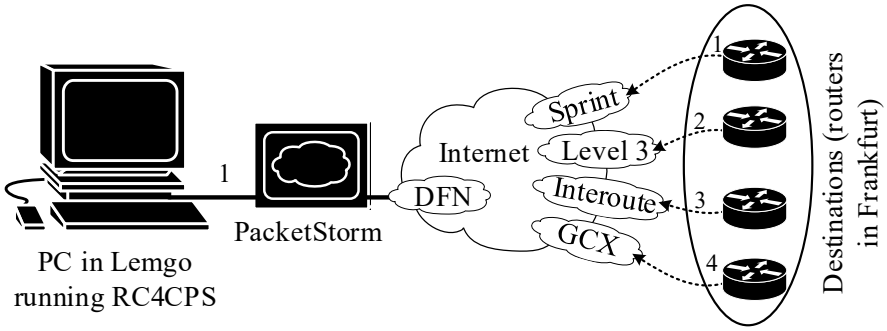


Figure 10.2 Evaluation setup for the MATLAB implementation of RC4CPS using a single-homed PC.

10.2.2 Results of Evaluation Setup 1

The evaluation in this setup started with the *IMMS* phase done from 2016-11-21, at 04:00 pm to 2016-11-25, at 09:40 am. In the *IMMS*, the statistical characteristics of the different paths were analyzed. The paths (*1*,*1*), (*1*,*2*), and (*1*,*3*) are characterized by isolated and single unavailability events with a very low frequency. By contrast, path (*1*,*4*) has both isolated and bursty occurrences of events. To determine the MC model to be used for each path for unavailability prediction, the procedure described in Section 7.2 is used. As all models achieved high *cc* value, paths (*1*,*1*), (*1*,*2*), and (*1*,*3*) are modeled using GM while path (*1*,*4*) is modeled using HMM.

In the period from 2016-12-05 to 2016-12-11, RC4CPS was run according to the setup in Figure 10.2. In Figure 10.3, u_{ij} values recorded by RC4CPS for the first 15×10^3 s are plotted. As illustrated, only the paths (*1*,*1*) and (*1*,*3*) have unavailability less than u_r during the considered interval. However, if the total evaluation interval is considered, u_{ij} for the paths (*1*,*1*), (*1*,*2*), (*1*,*3*), and (*1*,*4*) is 0.0001%, 0.0021%, 0.00007%, and 0.0461% correspondingly. Hence, none of the e2e paths can support the required u_r. By contrast, $u(\theta)$ for all 2-path subsets is 0% in the first 15×10^3 s and less than or equal to

10^{-6}% for the total evaluation interval. As θ_{pr} and θ_{ba} were always selected from the 2-path subsets in this evaluation setup, only subsets with two paths are considered in the remaining analysis.

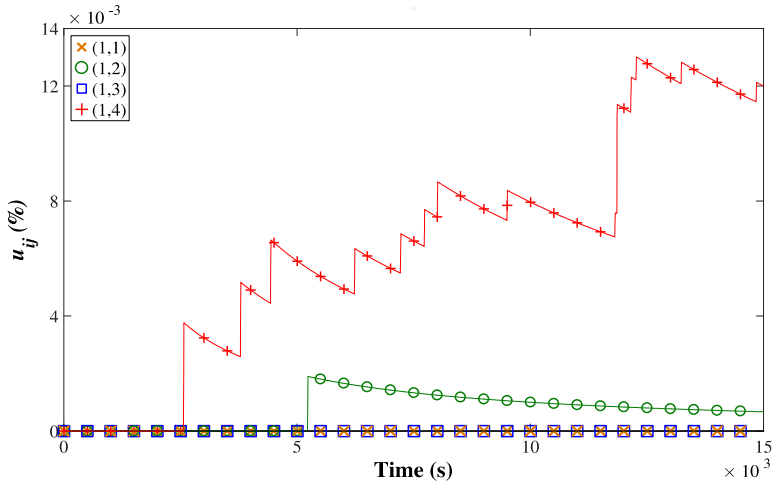


Figure 10.3 u_{ij} of e2e paths in the 1st evaluation.

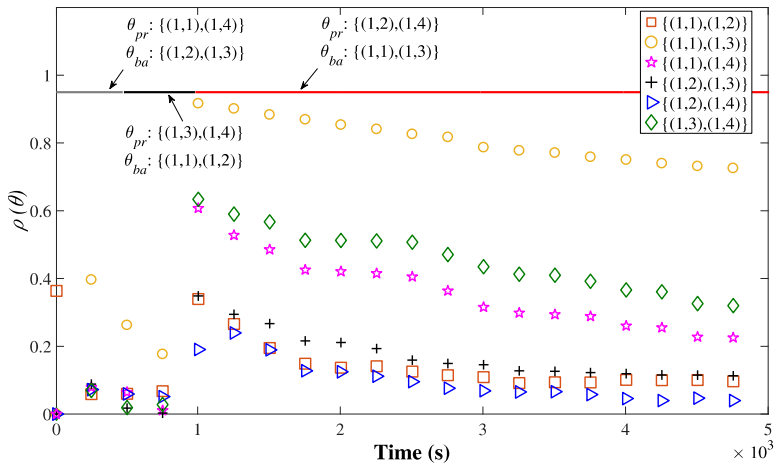


Figure 10.4 $\rho(\theta)$ for the 2-path subsets and the selected θ_{pr} and θ_{ba} during the first 5000 s in the 1st evaluation.

In Figure 10.4, $\rho(\theta)$ of the different 2-path subsets in the first 5×10^3 s is plotted. The selected subsets for θ_{pr} and θ_{ba} using the procedures described in Section 5.4 are indicated on the horizontal line at the top part of the figure. As it can be seen, the subset $\{(1,1),(1,4)\}$ is selected for θ_{pr} at the beginning of the considered interval. As the subset $\{(1,2),(1,3)\}$ has the minimum value for the summation in (5.13) among the subsets that have at least two other paths than those in θ_{pr}, RC4CPS selects it for θ_{ba}. In this evaluation, the periodic reselection for θ_{pr} and θ_{ba}, is done every 500 s. This interval was adequate to observe a significant increase of $\rho(\theta)$ in a similar experiment to that in Figure 10.6. Nevertheless, the periodic reselection of θ_{pr} and θ_{ba} is expected to be dependent on the characteristics of the monitored paths as well as the application requirements (Sections 5.4 and 7.3).

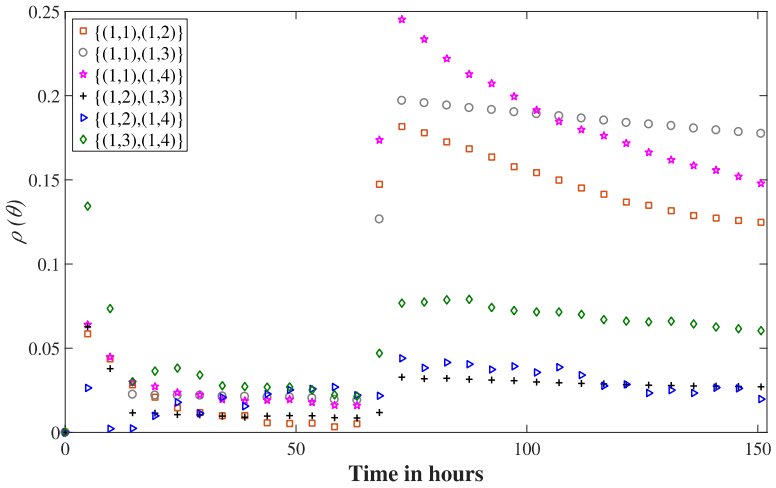


Figure 10.5 *ρ(θ)* for the 2-path subsets in the 1st evaluation.

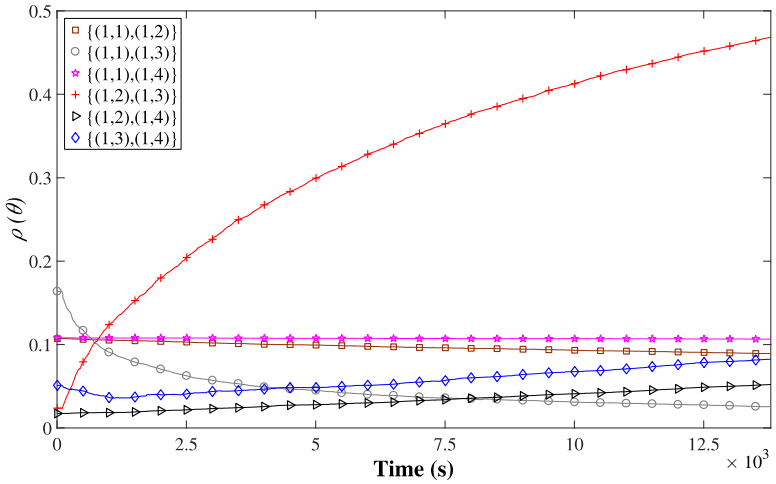


Figure 10.6 *ρ(θ)* for the 2-path subsets in the 1st evaluation after activating a shared random delay on the paths (*1*,*2*) and (*1*,*3*).

For the complete evaluation interval, *ρ(θ)* is shown in Figure 10.5 (values higher than 0.25 are not shown). An increase in *ρ(θ)* for all subsets in the middle of the evaluation interval can be observed. From this observation, it is expected that the packets belonging to the different e2e paths have experienced similar impacts over the shared links between these paths. To further investigate this observation, an exponential delay distribution with a mean of 50 ms (applied using the PacketStorm) was used to delay packets on the paths (*1*,*2*) and (*1*,*3*) that had the lowest value for *ρ(θ)*. The selection of this mean value is motivated by the measurements carried out in [154]. The results of the test are illustrated in Figure 10.6. The test was done on 2016-12-13 and all path logs (continuously logged data since 2016-12-05) were considered. As shown in the figure, the larger the number of packets that experience similar impacts, the higher the correlation. Consequently, it is expected that the larger the number of joint links between two paths, the higher the probability to observe high value for *ρ(θ)*.

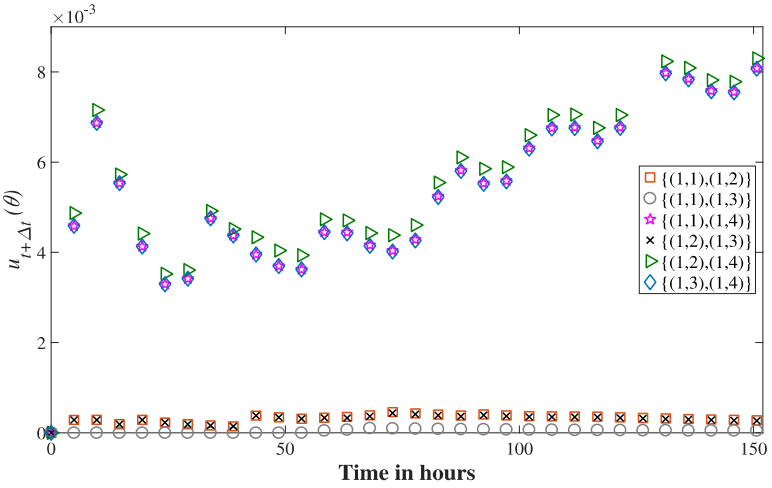


Figure 10.7 $u_{t+\Delta t}(\theta)$ for the 2-path subsets in the 1st evaluation.

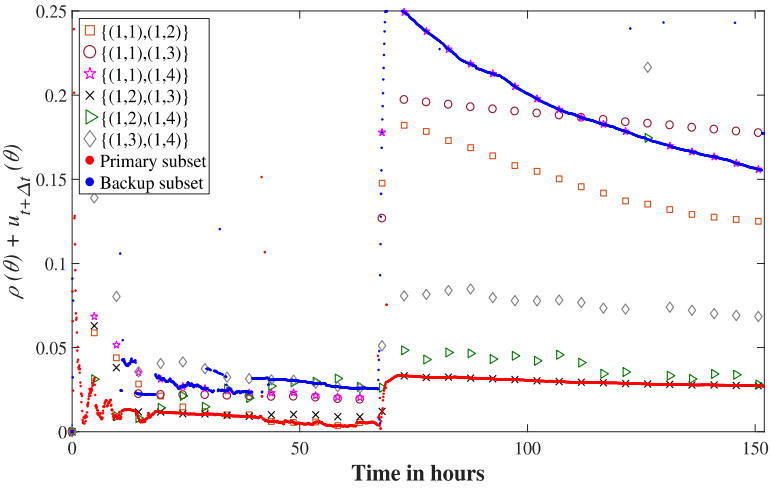


Figure 10.8 The sum of $\rho(\theta)$ and $u_{t+\Delta t}(\theta)$ for the 2-path subsets in the 1st evaluation.

By considering the evaluation interval from 2016-12-05 to 2016-12-11 again, Figure 10.7 shows $u_{t+\Delta t}(\theta)$) for the 2-path subsets. As it can be observed in the figure, $u_{t+\Delta t}(\theta)$) of the subsets that include the path (*1*,*4*) has an increasing trend. This is due to the increasing number of unavailability events indicated by the increasing unavailability of the path in Figure 10.3.

As given in (5.13), RC4CPS selection is based on the sum of $u_{t+\Delta t}(\theta)$ and $\rho(\theta)$. This sum is shown in Figure 10.8 for the whole evaluation interval. As illustrated in the figure, RC4CPS selects the first subset that fulfills the requirement on maximum unavailability and has the minimum sum of $u_{t+\Delta t}(\theta)$ and $\rho(\theta)$ for θ_{pr}. RC4CPS uses the same criteria to select θ_{ba} from the remaining subsets with preference for subsets that have at least two different paths from those used in θ_{pr}. For example, the selected subsets for θ_{pr} and θ_{ba} by RC4CPS after 100 hours are {(*1*,*2*),(*1*,*3*)} and {(*1*,*1*),(*1*,*4*)} respectively.

The evaluation results presented in this section demonstrate the observed selection metrics by RC4CPS that are calculated in an online manner. The results show also how RC4CPS perform the online selection of θ_{pr} and θ_{ba}.

10.2.3 Evaluation Setup 2

In the second evaluation setup, the ability of RC4CPS to recognize subsets with high diversity and low unavailability probability is further assisted using a multihomed PC. More specifically, RC4CPS is evaluated in two scenarios where e2e paths that traverse different hops/networks are available. The setup for this evaluation is shown in Figure 10.9. As illustrated, a multihomed PC with the MATLAB implementation of RC4CPS is used. The PC connects to two access ISPs using a wired and a cellular connections (through the Deutsche Telekom and the DFN networks) to reach the Internet. The location of the destination VN and its access networks are different for each of the two considered scenarios. These destination VNs will be described later in each scenario. The number assigned to each network interface is shown in Figure 10.9 and is the same in both scenarios.

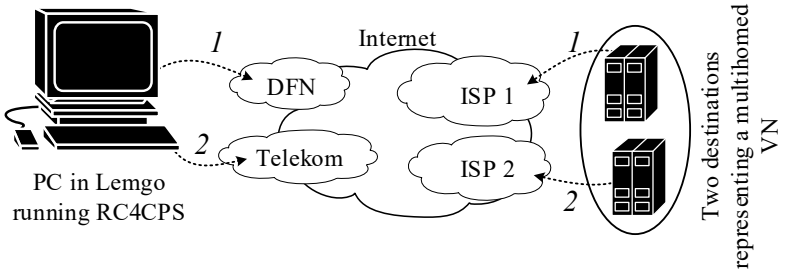


Figure 10.9 Evaluation setup for the MATLAB implementation of RC4CPS using a multihomed PC.

10.2.4 Scenarios and Results of Evaluation Setup 2

Scenario 1:

In this scenario, two end-systems in close geographical proximity (about 50 km apart) in *Italy* were selected to emulate a multihomed VN. The end-systems are part of the PlanetLab research network and are connected to the networks of the University of Parma and that of the University of Modena and Reggio Emilia. Both of these networks are part of the GARR network (acronyms' descriptions are provided in Table 6.1). The evaluation using this scenario was conducted in the interval from 2017-04-05, at 23:29 to 2017-04-06, at 17:35. From this interval, only the first 18 hours were considered for the results analysis. The considered e2e paths in this scenario are illustrated in Figure 10.10. The traversed networks and ASNs by the paths are provided in Table 10.1 and were inquired using the *Traceroute* tool. The number of shared hops (including the source and destination interfaces in the count) between the different e2e paths is provided in Table 10.2.

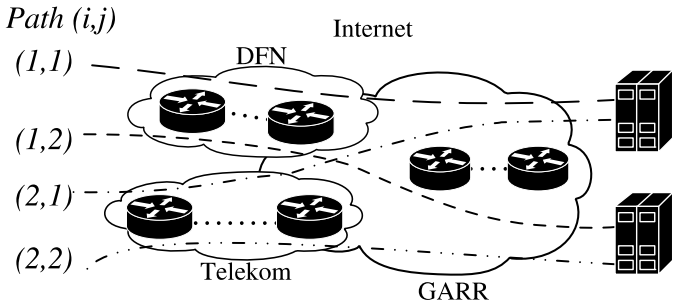


Figure 10.10 *Scenario 1* for the evaluation setup in Figure 10.9.

Table 10.1 Traversed ASNs and networks in *Scenario 1*.

Path	ASs	Networks
(*1*,*1*)	AS680, AS21320, AS137	DFN, GÉANT, GARR
(*1*,*2*)	AS680, AS21320, AS137	DFN, GÉANT, GARR
(*2*,*1*)	AS3320, AS9057, AS137	Telekom, Level3, GARR
(*2*,*2*)	AS3320, AS9057, AS137	Telekom, Level3, GARR

Table 10.2 Number of shared hops between the 2-path subsets in *Scenario 1*.

Subsets of 2 paths	# of Hops
(*1*,*1*),(*1*,*2*)	11
(*1*,*1*),(*2*,*1*)	6
(*1*,*1*),(*2*,*2*)	4
(*1*,*2*),(*2*,*1*)	0
(*1*,*2*),(*2*,*2*)	4
(*2*,*1*),(*2*,*2*)	11

In this scenario, $\rho(\theta)$ for the 2-path subsets is depicted in Figure 10.11 (only values less than 0.8 are shown in the figure). As illustrated, the subsets that share one source or one destination interface have high correlation. Even though that the subset {(*1,1*),(*2,1*)} had a very low value of $\rho(\theta)$ at the beginning of the evaluation interval, the value start to increase after 10 hours. Moreover, the subsets {(*1,1*),(*2,2*)} and {(*1,2*),(*2,1*)}have the lowest numbers of shared hops, namely 4 and 0. These two subsets have a very low value for $\rho(\theta)$ throughout the evaluation interval. Lastly, the subset {(*1,1*),(*1,2*)} has the lowest correlation compared to the other subsets. By comparing Figure 10.11 with Figures 10.5 and 10.18, it can be seen than $\rho(\theta)$ of the subset {(*1,1*),(*1,2*)} is not always very low. As descried in Section 7.4, this is attributed to the low utilization of the shared links between the two paths in the considered time interval.

$\rho(\theta)$ for the subsets that fulfill the last two conditions in (5.14) is provided in Figure 10.12. Values greater than 0.8 are not shown in the figure. In addition, θ_{pr} and θ_{ba} are also indicated in the figure. An important observation here is the change of the subset selected for θ_{ba} after 13 hours from the subset {(*1,2*),(*2,1*)} to {(*1,1*),(*1,2*),(*2,1*)}. This is because of the failure to fulfill the threshold u_r by the subset {(*1,2*),(*2,1*)}. With this regard, Figure 10.13 indicates which subsets of those considered in Figure 10.12 fulfill the first condition in (5.14), namely the unavailability threshold u_r. As indicated by the dotted horizontal line and after about 13 hours, $u(\theta)$ of the subsets {(*1,2*),(*2,1*)} and

$\{(1,2),(2,1),(2,2)\}$ becomes higher than u_r. Therefore, these subsets are not considered thereafter in the selection of θ_{pr} and θ_{ba} even though that the subset $\{(1,2),(2,1)\}$ has a lower correlation than the other subsets. It is necessary to indicate here that the lines connecting the data points in Figure 10.13 do not indicate the actual values between the data point and are used for illustration only.

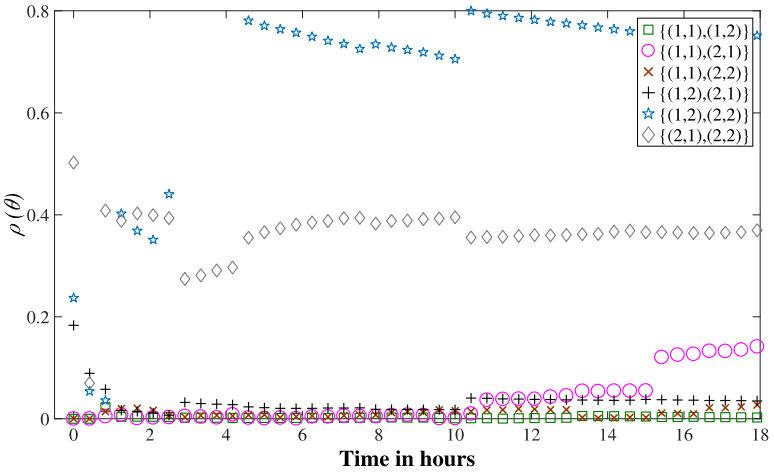


Figure 10.11 $\rho(\theta)$ of the 2-path subsets in the 1st *Scenario* of the 2nd evaluation.

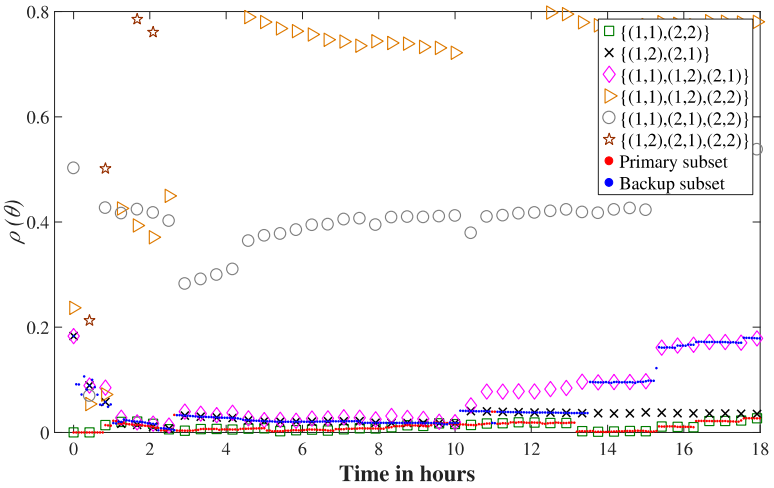


Figure 10.12 $\rho(\theta)$ for the 2- and 3-path subsets in the 1st *Scenario* of the 2nd evaluation.

The second metric in (5.13) is the MP unavailability probability, $u_{t+\Delta t}(\theta)$, which is given in Figure 10.14. Values greater than 0.05 are not shown in the figure for clarity of presentation. As illustrated, the path subsets selected for θ_{pr} and θ_{ba} have low values of $u_{t+\Delta t}(\theta)$. The change of the selected subset for θ_{ba} after 13 hours cannot be easily identified due to the clustered markers in the figure. $u_{t+\Delta t}(\theta)$ declined for all subsets in the first part of the figure as the number of considered samples in the unavailability prediction increased. On the other hand, $u_{t+\Delta t}(\theta)$ increased in the last part of the figure indicating an increase of unavailability events for some paths.

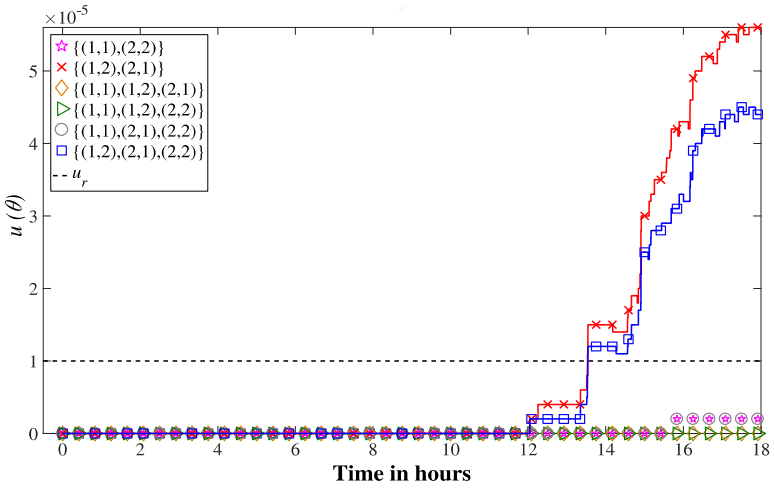


Figure 10.13 $u(\theta)$ for the 2- and 3-path subsets in the 1st *Scenario* of the 2nd evaluation.

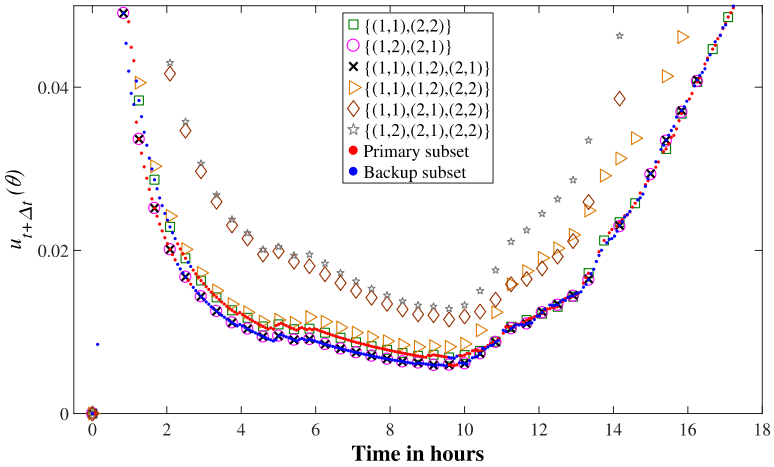


Figure 10.14 $u_{t+\Delta t}(\theta)$ for the 2- and 3-path subsets in the 1st *Scenario* of the 2nd evaluation.

Figure 10.15 illustrates only $u_{t+\Delta t}(\theta)$ for the subsets selected for θ_{pr} and θ_{ba}. As indicated previously, θ_{ba} was not the same subset during the evaluation interval. More specifically Figure 10.15 gives $u_{t+\Delta t}(\theta)$ for the selected subset for θ_{pr} and θ_{ba} at the considered instant of time. As illustrated in the figure, the values of $u_{t+\Delta t}(\theta)$ for θ_{pr} and θ_{ba} fluctuate between two levels. The lower of these represents $u_{t+\Delta t}(\theta)$ when the paths of each subset are available, namely when only p_{01} values in (5.8) are present in $u_{t+\Delta t}(\theta)$. By contrast, the higher level(s) of values of $u_{t+\Delta t}(\theta)$ represent(s) the case when one or more paths of the selected subsets for θ_{pr} or θ_{ba} experience unavailability events, namely when p_{11} values in (5.8) are present in $u_{t+\Delta t}(\theta)$. As shown in Figure 10.15, the first two thirds of the evaluation interval witness light occurrences of unavailability events only. This is indicated by the absence of high values of $u_{t+\Delta t}(\theta)$ in those parts compared to the third part of the evaluation interval.

Lastly, the sum of $\rho(\theta)$ and $u_{t+\Delta t}(\theta)$ of those subsets that fulfill the last two conditions in (5.14) is plotted in Figure 10.16. As shown, the selected subsets for θ_{pr} and θ_{ba} has the lowest sum in the figure. In addition, it can be seen that even though the subset

$\{(1,2),(2,1)\}$ has a lower sum than that of $\{(1,1),(1,2),(2,1)\}$, it is not selected as a backup subset after about 13 hours due to its high unavailability (as illustrated previously in Figure 10.13).

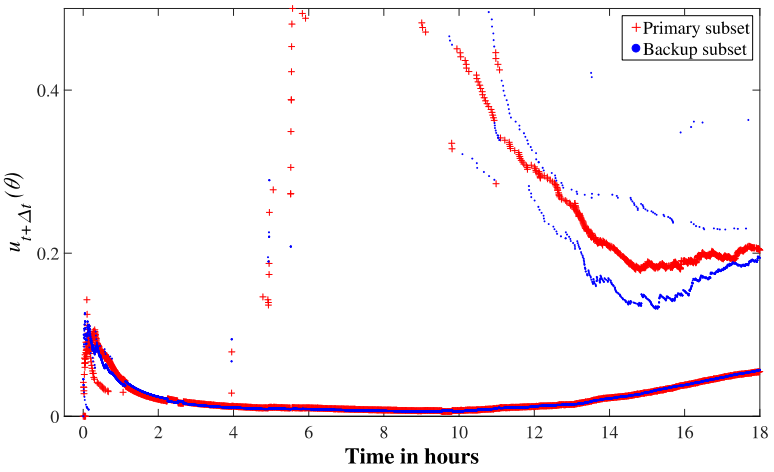


Figure 10.15 $u_{t+\Delta t}(\theta)$ of the subsets selected for θ_{pr} and θ_{ba} in the 1st *Scenario* of the 2nd evaluation.

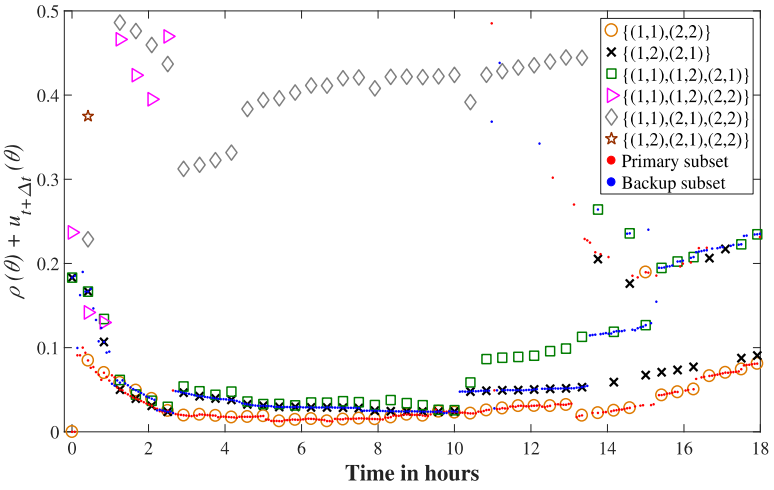


Figure 10.16 The sum of $\rho(\theta)$ and $u_{t+\Delta t}(\theta)$ for the 2- and 3-path subsets in the 1st *Scenario* of the 2nd evaluation.

Scenario 2:

In this scenario two DNS root servers are selected to represent the destination VN in Figure 10.9. The servers are located in *Frankfurt, Germany* and belong to the Cogent and the RIPE NCC networks. The resulting setup for this scenario and the considered e2e paths are illustrated in Figure 10.17. Tables 10.3 and 10.4 list the networks and ASNs traversed by the e2e paths and the number of shared hops by each 2-path subset. The information in these two tables was collected using the *Traceroute* tool. The conducted evaluation using this scenario started on 2017-04-16, at 19:49 and ended on

2017-04-17, at 23:05. For consistency of results presentation in the *Scenarios 1* and *2*, only the data collected in the first 18 hours of the evaluation are presented.

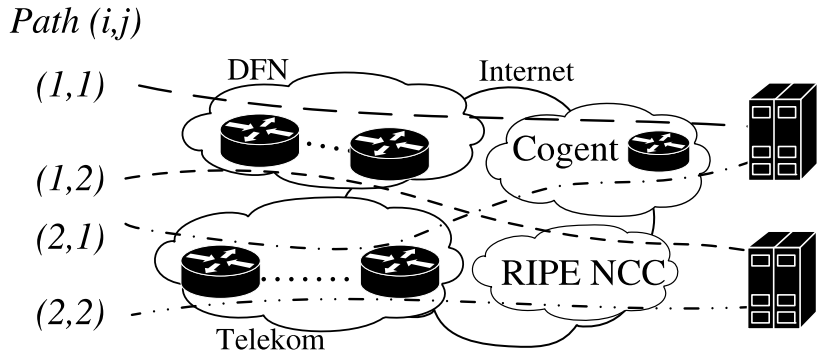


Figure 10.17 *Scenario 2* for the evaluation setup in Figure 10.9.

Table 10.3 Traversed ASNs and networks in *Scenario 2*.

Path	ASs	Networks
(*1*,*1*)	AS680, AS2149	DFN, Cogent
(*1*,*2*)	AS680, AS25152	DFN, RIPE NCC
(*2*,*1*)	AS3320, AS2149	Telekom, Cogent
(*2*,*2*)	AS3320, AS25152	Telekom, RIPE NCC

Table 10.4 Number of shared hops between 2-path subsets in *Scenario 2*.

Subsets of 2 paths	# of Hops
(*1*,*1*),(*1*,*2*)	6
(*1*,*1*),(*2*,*1*)	2
(*1*,*1*),(*2*,*2*)	0
(*1*,*2*),(*2*,*1*)	0
(*1*,*2*),(*2*,*2*)	1
(*2*,*1*),(*2*,*2*)	10

As illustrated in Figure 10.17 and Table 10.4, only the subsets that share the same source interface have a large number of shared hops. On the other hand, the 2-path subsets that end at the destination interface connected to the Cogent network have one shared hop. Moreover, the subsets {(*1,1*),(*2,2*)} and {(*1,2*),(*2,1*)} share no hops. Consequently, the correlation of the 2-path subsets with a shared source interface is expected to be higher than the other subsets. This can be seen in Figure 10.18 that shows $\rho(\theta)$ for all 2-path subsets.

If only those subsets that satisfy the last two conditions in (5.14) are considered, then Figure 10.19 is obtained. As Figures 10.18 and 10.19 show, the 2-path subsets, which traverse different networks, have the lowest values for $\rho(\theta)$.

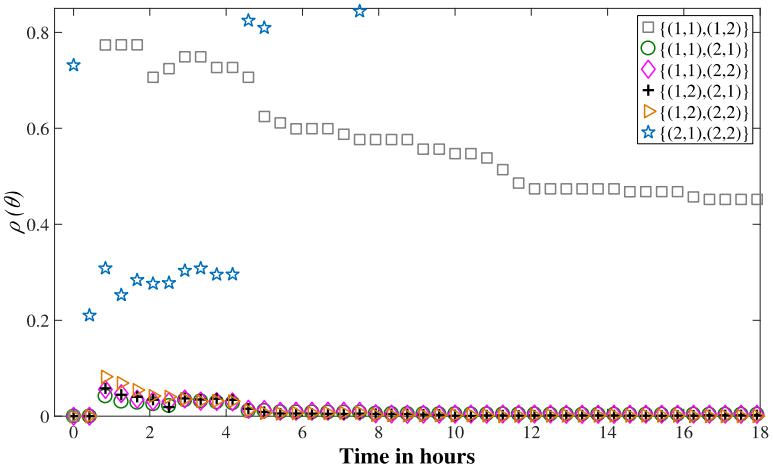


Figure 10.18 *ρ(θ)* of the 2-path subsets in the 2nd *Scenario* of the 2nd evaluation.

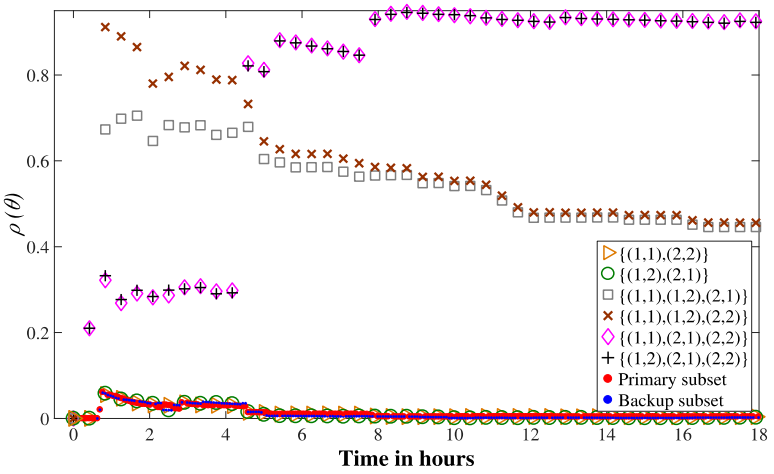


Figure 10.19 *ρ(θ)* for the 2- and 3-path subsets in the 2nd *Scenario* of the 2nd evaluation.

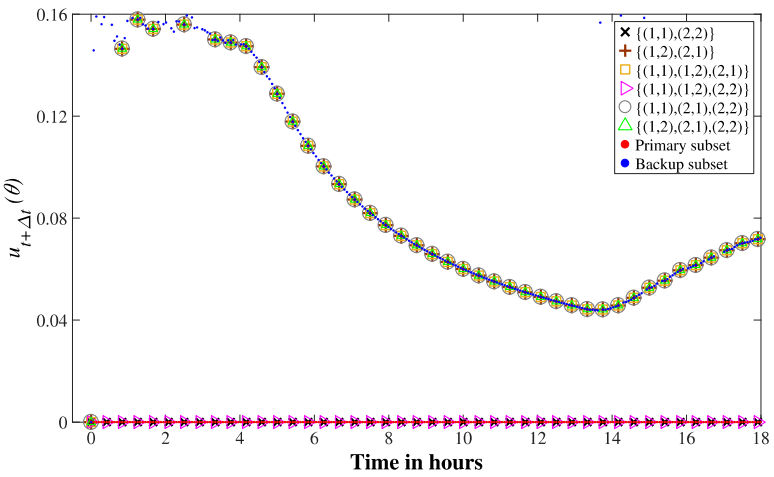


Figure 10.20 $u_{t+\Delta t}(\theta)$ for the 2- and 3-path subsets in the 2nd *Scenario* of the 2nd evaluation.

$u_{t+\Delta t}(\theta)$ for the subsets considered in Figure 10.19 is provided in Figure 10.20. As illustrated, The subset $\{(1,2),(2,1)\}$ as well as the subsets that include its paths have higher values for $u_{t+\Delta t}(\theta)$.This is mainly due to the frequent unavailability events on the path $(2,1)$ as it can be observed from u_{ij} values of the individual paths in Figure 10.21. Nevertheless, $u(\theta)$ for all 2- and 3-path subsets with the last two conditions in (5.14) fulfilled was zero throughout the considered evaluation interval.

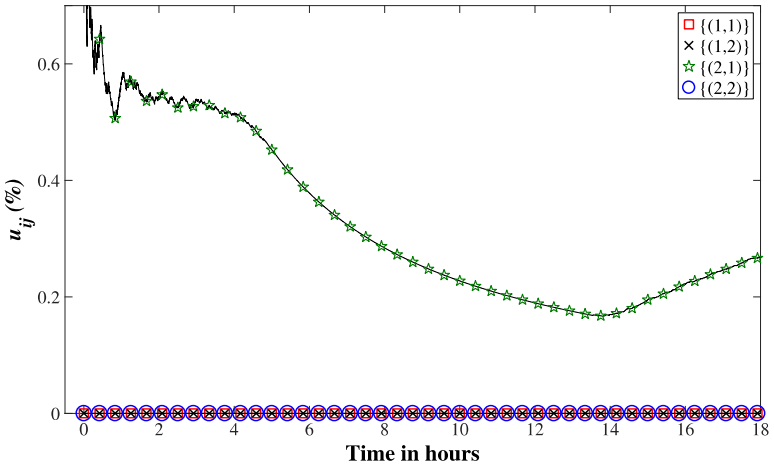


Figure 10.21 u_{ij} of e2e paths in the 2nd *Scenario* of the 2nd evaluation.

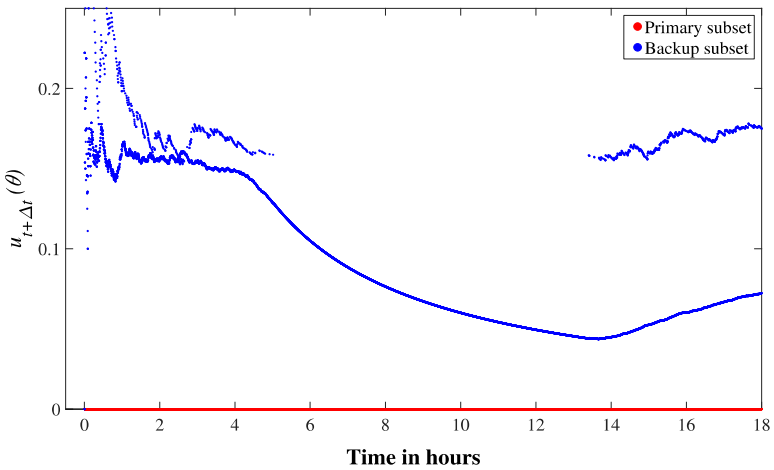


Figure 10.22 $u_{t+\Delta t}(\theta)$ of the subsets selected for θ_{pr} and θ_{ba} in the 2nd *Scenario* of the 2nd evaluation.

To observe the impact of the presence of unavailability events on $u_{t+\Delta t}(\theta)$, Figure 10.22 is depicted. In the figure, $u_{t+\Delta t}(\theta)$ for the subsets selected for θ_{pr} is zero. This indicates that none of the selected subsets for θ_{pr} included the path $(2,1)$ that has frequent unavailability events. RC4CPS selection for θ_{ba} prefers the subsets that fulfill the conditions in (5.14) and have at least two other paths than those in θ_{pr}. All candidates in this case include the path $(2,1)$. As it can be seen in Figure 10.22, $u_{t+\Delta t}(\theta)$ for the subsets

selected for θ_{ba} fluctuates between two levels at the first and third portions of the evaluation interval where the path (*2*,*1*) experienced frequent unavailability events.

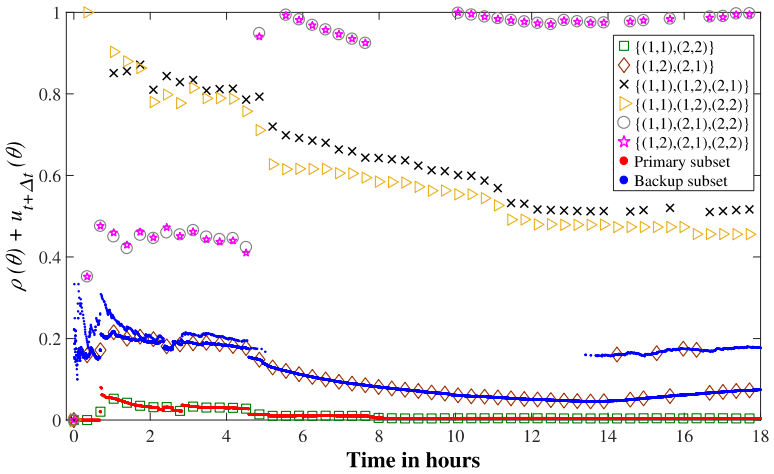


Figure 10.23 The sum of $\rho(\theta)$ and $u_{t+\Delta t}(\theta)$ for the 2- and 3-path subsets in the 2nd *Scenario* of the 2nd evaluation.

Lastly, Figure 10.23 plots the sum $\rho(\theta) + u_{t+\Delta t}(\theta)$ for the subsets with two and three paths that satisfy the last two conditions in (5.14). The figure shows that the subset with the minimum sum as given in (5.13) is selected for θ_{pr}, namely the subset $\{(1,1),(2,2)\}$. As the subset $\{(1,2),(2,1)\}$ has the minimum sum of $\rho(\theta)$ and $u_{t+\Delta t}(\theta)$ among the subsets with two other paths than those in θ_{pr}, it is selected for θ_{ba}.

The results from the 2nd evaluation setup show clearly the effectiveness of the used selection metrics in determining the subsets with the highest diversity and the lowest unavailability among their paths. The results show also the online calculation of the selection metrics and demonstrate the online decision-making procedures of RC4CPS (Section 5.4) to provide dynamic MP selection. Lastly, the results demonstrate the ability of RC4CPS to support the high reliability requirements of smart grids.

10.3 Discussion

The evaluations results presented in this chapter provide a number of observation regarding the utilized mechanisms and the main functions of the RC4CPS approach. These observations are mainly drawn from the second evaluation setup. This is due the multihomed nature of the source node. In the following, a brief discussion regarding these observations is provided.

10.3.1 MP Diversity Estimation

The main observation regarding the diversity estimation mechanism is as follows. e2e paths that do not share hops have very low values for $\rho(\theta)$. On the other hand, e2e paths that share one or more hops such as those that share an interface (at the source or at the destination) are likely to have higher values for $\rho(\theta)$, especially, over long time

intervals. For example, $\rho(\theta)$ for the subsets {(*1,1*),(*2,2*)} and {(*1,2*),(*2,1*)} have very low values in the two scenarios of the 2nd evaluation setup. On the other hand, $\rho(\theta)$ for the subsets {(*1,1*),(*1,2*)} and {(*2,1*),(*2,2*)} have higher values than the other subsets in at least one of the scenarios. Here, I refer to Figure 10.5 in the first evaluation setup where all the 2-path subsets had an increase near the mid of the evaluation interval. This emphasize the conclusion that a subset of e2e paths that share a number of hops might not have a high correlation over a short time interval, but rather, tend to have high correlation over long time intervals. This might be attributed to the low utilization of the shared links and hops as indicated in Section 7.4.

10.3.2 MP Unavailability Prediction

When unavailability events occur on one or more of the paths in a subset, $u_{t+\Delta t}(\theta)$ fluctuates between two or more levels of values. The lower one of these is associated with unavailability probabilities where all paths are available, namely the p_{01} values of (5.8). The higher ones are associated with unavailability probabilities where one or more paths are unavailable, namely one or more of the p_{11} values of (5.8). The $u_{t+\Delta t}(\theta)$ levels are not fixed and change based on the number of unavailability events and if it is increasing or decreasing over time (Section 7.1). Hence, the more available are the paths, the lower the level(s) of $u_{t+\Delta t}(\theta)$ values. For example, if $u_{t+\Delta t}(\theta)$ for θ_{pr} and θ_{ba} in *Scenario 2* (Figure 10.22) is considered, then it is clear that the paths of the subsets selected for θ_{ba} have more and frequent unavailability events as indicated by the presence of multiple levels for the values for $u_{t+\Delta t}(\theta_{ba})$. These observations show how $u_{t+\Delta t}(\theta)$ reflects the state of the individual paths of a subset and emphasize the importance of including $u_{t+\Delta t}(\theta)$ in the MP selection.

11 Implementation and Evaluation of RC4CPS Using iPRP MP Transport Protocol

11.1 Introduction

iPRP is a MP transport protocol based on UDP [155]. It was developed as part of a PhD thesis [156] using the concept of PRP [18]. The protocol provides high reliability by means of redundant MP communication in IP networks for smart grids. More specifically, multiple e2e paths are used to send duplicated packets of chosen UDP flows. In this case, lost packets on one path are compensated by their copies from the other paths. The communication using iPRP continues even if one path failed, but with degraded-redundancy. The e2e paths are attained through multiple interfaces that are connected through disjoint networks (physically or logically). Hence, for iPRP to work optimally, control over the networks infrastructure is needed.

There are two implementations of the iPRP design that belongs to two different projects. The first implementation is based on IPv6 while the second implementation is based on IPv4 [147]. Although the IPv4-based implementation is a work in progress, many features from the original iPRP specifications in [156] are already implemented. This section presents the unicast mode of iPRP using its specifications given in [156]. The multicast mode of iPRP is not considered due to the unicast nature of RC4CPS. Nevertheless, extending RC4CPS and its iPRP implementation is left for future development of the approach.

The protocol offers a number of desired features including providing high reliability, reducing e2e time delay, compatibility with middleboxes, and compatibility with existing applications (transparent to both the application and network layers). In addition, the deployment of iPRP requires only multihomed end-systems and a few configuration steps. These include enabling its control and data ports and selecting the application ports for which iPRP should be used. The control port is used for exchanging information related to connection maintenance while the data port is utilized for receiving and sending the duplicated data. After successful installation of iPRP, the receiving device will listen to the monitored application ports. Once a UDP packet arrives at one of the monitored ports and the sender is iPRP capable, an iPRP session is initiated. Any further UDP packets targeting the said monitored port will now be replicated by the sender and transmitted via multiple predefined interfaces, using the iPRP data port. The receiver will forward the first received copy of each packet to the application and discard other duplicates.

11.2 Protocol Description

The iPRP design was planned based on conditions and requirements drawn from the application area of smart grids. The protocol was designed for WANs with controlled infrastructure. The design considered minimizing the deployment efforts of the protocol. With this regard, iPRP work on top of the legacy UDP protocol to ensure compatibility with middleboxes. In addition, to provide MP communication without modifying network equipment, iPRP relies on using multiple interfaces with the requirement that they are connected to disjoint networks. Disjoint paths are provided if the used networks are separated physically (Figure 11.1a) or logically. For the later case, interconnected networks are divided into different logical networks or sub-clouds. This facilitates network management and results in easier accessibility and controllability (Figure 11.1b). For iPRP to function in such an environment, network arrangements are needed such that interconnecting links between sub-clouds as those shown in Figure 11.1b route no traffic between the sub-clouds if it is sent to a destination within one cloud. In other words, only traffic targeting destinations in a different sub-cloud should cross the boundaries over interconnecting links between the logical networks.

The number of physical or logical networks is the same as the number of e2e paths that can be established by iPRP. Consequently, the used paths for iPRP should be fail-independent to attain the maximum benefits of MP communication. iPRP sends copies from the original packet using the available e2e paths and removes the not needed duplicates at the receiver. The e2e delay experienced by end-systems is basically the delay experienced by the first copy of a packet to arrive. Hence, iPRP usage also reduces transmission delays of the individual application packets. By contrast, iPRP focuses not on recovering from transmission errors and not on preventing them. In addition, iPRP and unlike TCP, does not offer congestion control. Nevertheless, it maintains connections during transient transmission problems but without solving them.

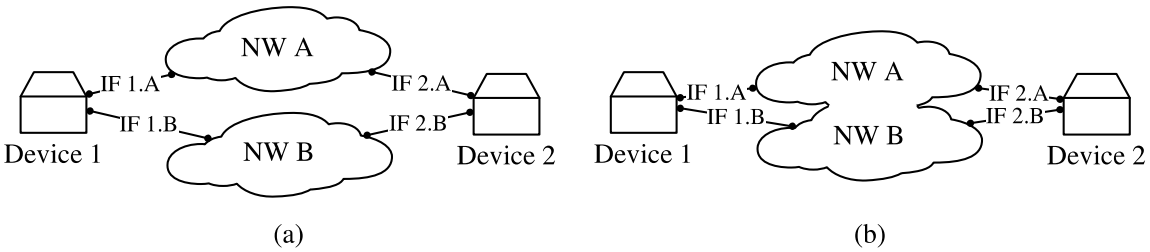


Figure 11.1 (a) Multihomed devices connected to two physically separated networks A and B, (b) Multihomed end-systems connected to a network with two logical networks (sub-clouds) A and B.

As mentioned previously, iPRP duplicates only selected UDP flows. This is mainly because duplicating all data is not desired as it could induce network congestion and waste network resources.

11.2.1 Path Matching

iPRP requires non-joint paths. To ensure this, iPRP uses an identifier for each interface called iPRP Network subcloud Discriminator (IND). The authors of iPRP suggest computing it using the interface's IPs or their fully qualified domain names. During iPRP sessions initiation, the available INDs at the receiver are advertised to the sender side to compare them with its own INDs. When a pair of matching INDs is detected, the sender side creates a peer-base entry with the respective sender and receiver IPs stored. That is, for each peer-base entry there is a corresponding iPRP session. In this case, if a receiver has a peer-base entry, then all stored IPs (paths) are used to replicate transmitted data.

11.2.2 Protocol Function Blocks

The iPRP protocol is divided into two planes, the control and the data planes. The control plane establishes and maintains a connection while the data plane is responsible for replicating, transmitting and discarding iPRP data packets. Once a UDP packet is received on a monitored port at the receiver, an iPRP session is initiated. A session is established throughout several steps, handled by the function blocks at the sender and receiver sides. A sequence chart that describes the iPRP initialization and the interaction of the function blocks is shown in Figure 11.2. As illustrated, the control plan on the receiver side includes the Soft-state-maintenance and the iPRP-capability-advertisement blocks. On the sender side, it includes the iPRP-session-maintenance block. The data plan on the receiver side has the Duplicate-discard block while on the sender side it has the Packet-replication block. In the following, a brief description about these blocks is provided (All referenced algorithms are provided in Appendix C).

Control plane:

1. *Soft-state-maintenance block* (Alg. C-4) (Receiver)

When a monitored port by iPRP receives a UDP packet with unknown sender information, then the IP address of the sender, the source port number, and the destination port number of the received packet are stored. These are stored in a list that contains all active senders. When the sender information is already known, then the last-seen timer for the sender is updated. Once the last-seen timer of a sender expires, which happens when no messages are received on the monitored port for a specific period of time, the sender gets removed. UDP packets arriving at other ports (non-monitored ports) will not trigger iPRP.

2. *iPRP-capability-advertisement block* (Alg. C-5) (Receiver)

The iPRP receiver side broadcasts session relevant information every T_{CAP} using the capability messages (*iPRP_CAP*) via the control port to its active senders. These messages contain the following information: (i) iPRP capability; (ii) iPRP version; (iii) the receiver's INDs; (iv) the receiver IP addresses; and (v) the port numbers for the

UDP packet which triggered the message such that the sender can recognize the correct session and associate the information to it.

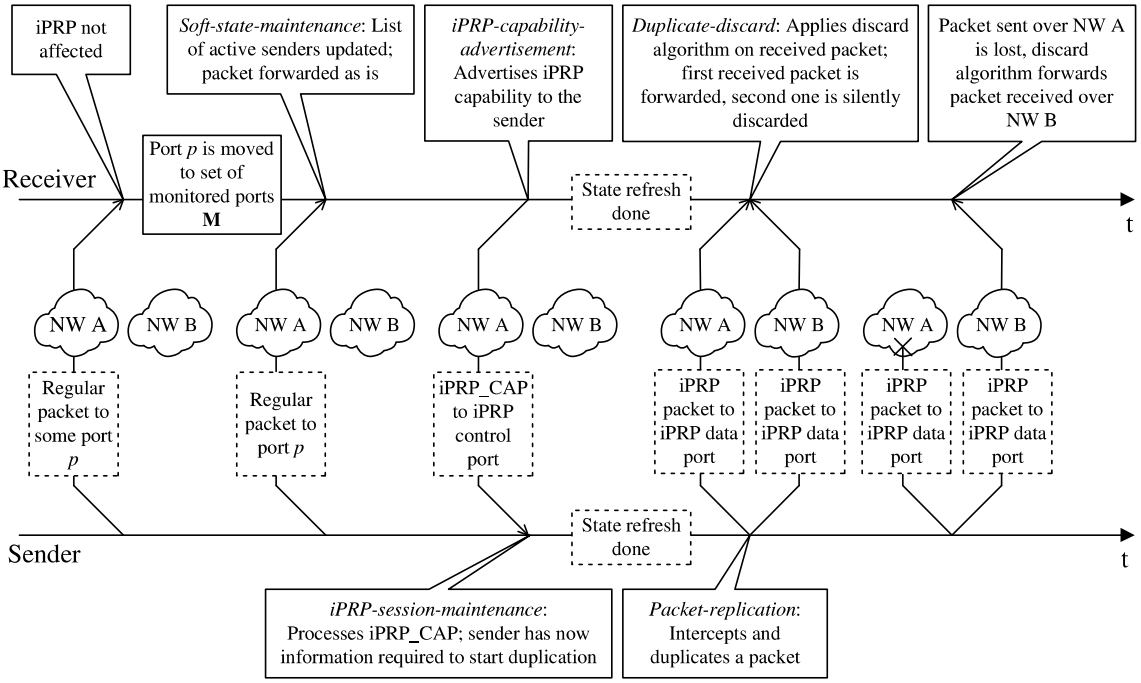


Figure 11.2 The sequence chart of how an iPRP session starts between iPRP-capable end-systems over two networks A and B [156].

3. *iPRP session-maintenance block* (Alg. C-6) (Sender)

Once an *iPRP_CAP* message is received, the iPRP sender side extracts the receiver session information and the IND matching process is initiated. In the case of successful detection of matching INDs pair, a peer-base entry for the receiver is created. When the *iPRP_CAP* message belongs to an already existing session, then the iPRP sender updates the last-seen timer. If the last-seen timer expired without receiving *iPRP_CAP* message, the peer-base entry gets removed.

Data Plane:

4. *Packet-replication block* (Alg. C-7) (Sender)

The iPRP sender side intercepts each outgoing UDP packet and compares its receiver with the peer-bases. If the sender detected an active corresponding session for that specific destination, then the packet is further processed. This further processing includes replicating the packet's payload, prepending the replicates with iPRP headers, and transmitting them via the data port for iPRP and using the matched paths.

5. *Duplicate-discard block* (Alg. C-8 and Alg. C-9) (Receiver)

iPRP receiver side intercepts each UDP packet arriving at the data port of iPRP. It first checks for the sequence number to check if the packet is the first copy to be seen (fresh) or not. Then, the first copy of each packet is reconstructed to restore its original format

and forwarded to its corresponding application. The packet reconstruction is done using the information in the iPRP header. Any following copies of that particular packet will be silently discarded. Fresh packets that arrive later than subsequent packets are handled separately. Such packets are forwarded only if timeout threshold is not exceeded.

11.2.3 iPRP Header

Each packet in iPRP is appended with its own header located after the innermost UDP header. This allows iPRP to work as expected even when tunneling mechanisms are present. The structure of the header can be seen in Figure 11.3. The iPRP version allows the receiver to identify versions mismatch and notify the user of rejected incoming packets.

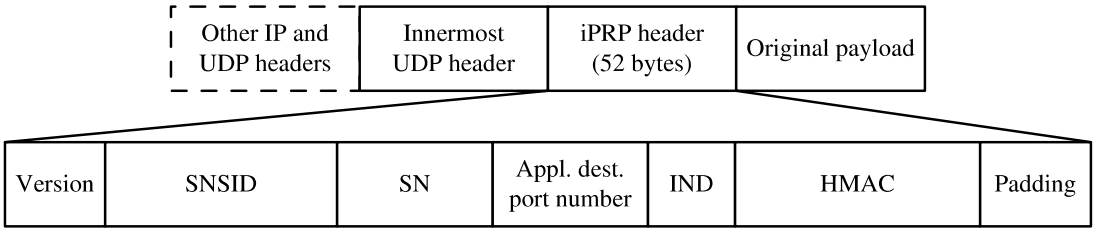


Figure 11.3 The location of iPRP header and its fields [17].

The Sequence-Number-Space ID (SNSID) identifies the individual iPRP sessions communicating with the same receiver. This way the receiver is able to recognize multiple iPRP sessions. It contains the source IP and UDP port and a reboot counter. In case the sender crashed and rebooted, the changed reboot counter will indicate a new SNSID. Nevertheless, the receiver can act upon the change and recover the session seamlessly. The Sequence Number (SN) is used to identify duplicate packets at the receiver where all duplicates are assigned the same SN. The original destination port number is also included to correctly form and forward the received packet to its targeted application. In addition, the currently utilized path IND is also included for debugging purposes.

11.2.4 Duplicate Discard Mechanism

Data duplication is done at the sender. The duplication targets outgoing UDP packet with a corresponding iPRP session which is associated with the destination socket. The payload of such packet is replicated on each of the used paths with rewritten UDP/IP header information. The constructed iPRP/IP header will contain the source and destination IPs and the iPRP data port as the destination port. At the receiver side, a discard algorithm is used to handle incoming packets and to determine whether they are the first copies to arrive and whether they are late. If the packet is the first to arrive among the set of its duplicates, then it is forwarded to the corresponding application. Otherwise, the packet is discarded.

11.3 iPRP implementation

The IPv4-based implementation of iPRP runs from the userspace in Linux and is written using the C programming language. It is available on GitHub [147] and for this work the commit from the 2016-11-28 was used. Kernel-relevant services such as socket creation/utilization, process control and intercommunication, and file management are handled using system calls. As mentioned previously, this implementation is a work in progress and lacks some of the features in the specifications in [156]. These missing features are briefly mentioned hereafter.

The security considerations in the used implementation are not complete yet. As a result, attackers in compromised networks can intercept iPRP packets and alter their sequence-number field. This way, the iPRP discard algorithm will drop these packets with old sequence numbers. To address such possibility, iPRP specification suggests encrypting the iPRP header and authenticating it using a key. The key is pre-shared and periodically updated. In addition, the specification suggests utilizing a datagram transport layer security session (DTLS) for the exchange of control messages. In the used implementation, the pre-shared key is checked and exchanged. However, the key is neither periodically updated nor encrypted. Moreover, it does not utilize a DTLS channel for exchanging the control messages.

The cleanup routines which delete relevant information for expired session at the sender and receiver sides are also not complete. With this regard, a mechanism to delete old peer-bases and to execute an orderly shutdown for the associated sender daemon is still missing.

The authors of iPRP also suggested a diagnostics toolkit. The toolkit shall exploit the TCP/IP diagnostic features and add to them the iPRP-specific tools. These tools should provide connectivity testing between iPRP communicating parties and acquire and print the sender and receiver statistics. Such toolkit is not implemented yet in the utilized iPRP implementation.

In iPRP specifications, INDs assignment should happen automatically with calculation following predefined rules (e.g. using IP addresses). However, the considered iPRP implementation simply utilizes incrementing integers. For an end-system with n interfaces, the assigned INDs will be sequentially assigned from 0 – n. The first interface entered to be utilized by iPRP is assigned 0, the next will be assigned 1, and so on. With such assignment, the input order of interfaces to be utilized by iPRP controls at which interfaces the utilized e2e paths will terminate.

11.3.1 Architecture

Figure 11.4 shows the general structure for the utilized implementation of iPRP. As illustrated, the implementation consists of four daemons where each daemon launches and maintains a number of threads for its own. These are the iPRP control daemon

(ICD), iPRP monitoring daemon (IMD), iPRP receiver daemon (IRD) and iPRP sender daemon (ISD). In Figure 11.4, the functions at the receiver side are always launched while those at the sender side are launched once the executing end-system participates as a sender in an iPRP-session. These four daemons can be considered as implementations of the individual function blocks of iPRP presented in Section 11.2.2. The main functions of each of the four daemons and the associated threads are explained below.

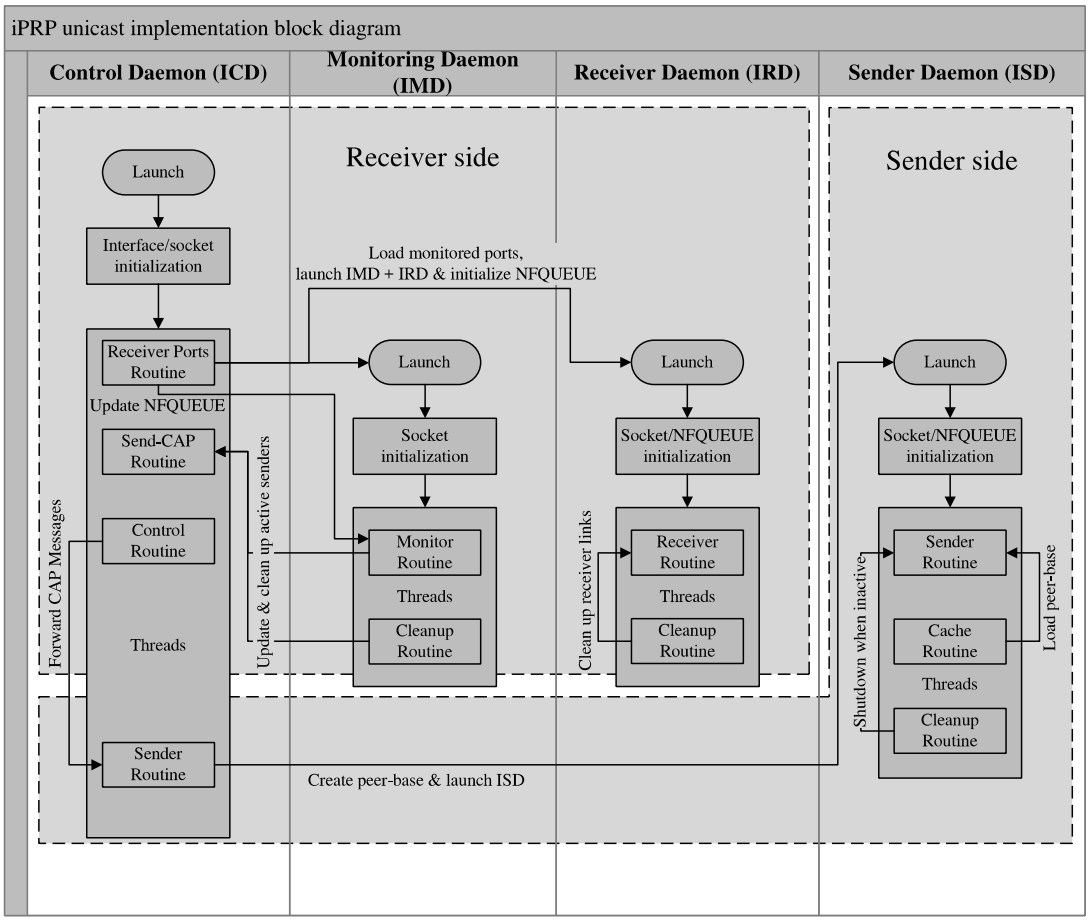


Figure 11.4 Architecture of iPRP implementation.

1. *iPRP control daemon (ICD)*

Receiver side:

The main entry in the iPRP implementation is the ICD. This daemon is responsible on processing the IP addresses of the end-system interfaces. These IPs are passed to iPRP as function arguments. In addition, the ICD is in charge of managing the other iPRP daemons. *The Receiver Ports Routine* of the ICD loads the ports which iPRP should monitor (provided in a configuration file for it). The routine also sets up the NFQUEUE rules (see Section 11.3.3) to be used by the IMD and IRD before it launches them. The IMD NFQUEUE rules are related to the monitored ports provided by the configuration file. If the ports were modified in the configuration file, then the routine updates the

IMD NFQUEUE rules to apply the changes. The iPRP-capability advertisement block that was introduced in Section 11.2.2 (described by Alg. C-5), is realized using the *Send-CAP Routine*. This routine periodically transmits *iPRP_CAP* messages that target all active senders. The *Control Routine* in the ICD waits for incoming *iPRP_CAP* messages which are forwarded after that to the *Sender Routine.* The *Control Routine* performs the functions of the iPRP session-maintenance block (described by Alg. A.6).

Sender side:

The *Sender Routine* in the ICD waits for *iPRP_CAP* messages. Once a message is received, it either creates a peer-base entry and launches the ISD when the receiver information is unknown or updates the expiration timer for the receiver in the peer-base when its information is known. When there are multiple ports monitored which belong to different applications, the ICD creates individual ISDs for each iPRP session for the ports, each with an own peer-base.

2. *iPRP monitoring daemon (IMD)*

The IMD represents the soft-state-maintenance block (described by Alg. C-4). It is responsible on monitoring the ports of the applications with the data to be duplicated using NFQUEUE rules. The list of active senders who are communicating with the end-system is managed by the IMD. A new entry is added to the list once a new sender starts sending packets on one of the monitored ports. The *Cleanup Routine* on the other hand removes entries that belong to senders who became inactive.

3. *iPRP receiver daemon (IRD)*

Receiver side:

The IRD represents the duplicate-discard block (described by Alg. C-8 and Alg. C-9). The daemon also uses the NFQUEUE rules to intercepts the packets from the iPRP data port. In addition, the maintenance of the receiver links (described in Section 11.3.2) for each iPRP session is done by this daemon. Intercepted packets are rebuilt to their original state before being forwarded to their corresponding applications. Expired receiver links are removed using The *Cleanup Routine*.

4. *iPRP sender daemon (ISD)*

Sender side:

The ISD performs the function of the packet-replication block (described by Alg. C-7). The daemon first loads the current peer-base through the *Cache Routine*. The *Sender Routine* of the daemon intercepts any data packets from the application layer which target a destination in peer-base. The interception is done as in other daemons using the NFQUEUE rules. The intercepted packets are then replicated and sent using the matched paths available in the peer-base. For expired peer-bases, the ISDs are terminated using the *Cleanup Routine.*

11.3.2 Session-information links and structures

Various data structures are used in the iPRP implementation. These data structures referred to using the terms *link* or *list.* They store session-relevant information and facilitate linking to the different iPRP sessions. As these data structures will be mentioned often in this chapter, a brief description about their purposes and contents is briefly provided below.

1. *Receiver link*

The receiver link is maintained by the IRD and represents the reference to an iPRP session at the receiver-side. It contains the following information: (i) the IPs of the sender, (ii) the application port of the sender, (iii) the SNSID, (iv) the last sequence number seen in order to handle duplicates, (v) a sequence number list which is used to handle delayed duplicates, and (vi) an expiration timer. The IRD utilizes the receiver link data structure in the duplicate-discard algorithm to handle iPRP sessions associated with different senders. The receiver link of a certain sender is deleted by the IRD when data duplication is stopped by the sender.

2. *Sender link*

The sender link is maintained by the ICD and refers to an iPRP session at the sender-side. It stores the following information: (i) the IP address of the receiver targeted by the application in the upper layer, (ii) the source and destination ports of the application data packet, (iii) the ID of the queue used by the associated NFQUEUE, and (iv) an expiration timer. The sender link itself is stored within a peer-base and is utilized to identify the corresponding receiver by the ISD. Once the receiver stops transmitting iPRP-CAP messages, the sender link is deleted by the *Cleanup routine* of the ISD.

3. *Active sender list*

The active sender list is maintained by the IMD to track all active senders. An update of the list is triggered by the arrival of a UDP packet on a monitored port by iPRP such that it is the first to arrive. This arrival of the first packet represents the first instance of initiating an iPRP session. Each entry in the active sender list stores the interface IP address of the source and the source and destination port numbers of the data packet which initiated the iPRP session. After maintaining the active sender list, an *iPRP_CAP* message is sent to each sender in the list.

4. *Peer-base*

The Peer-bases store the matched paths and the associated sender links. They are utilized by the ISD to set up and use the sockets for data replication.

11.3.3 Packet handling

The manipulation of the UDP protocol at the transport layer can be considered as the most important part of iPRP implementation. For iPRP to function, it is required that the

incoming data packets from the application layer be intercepted and changed transparently. For this purpose, the NFQUEUE (*libnetfilter_queue*) framework [157] is utilized. NFQUEUE is an *iptables* target that represents a userspace library such that an application programming interface (API) to queued packets by the kernel packet filter is provided. The API allows the delegation of decisions on the packets queued to a software at the userspace which, in this case, is iPRP. This enables iPRP to create rules to intercept packets, manipulate their content, and issue verdicts on them. The *libnfnetlink* library as well as a *nfnetlink_queue* compatible kernel are required to use NFQUEUE. Here, the message protocol between the userspace and the kernel communication sockets is the *libnfnetlink* library. The protocol is used to exchange verdicts, payloads as well as information about en-queued packets. Linux kernel versions above 2.6.14 are expected to support the *nfnetlink_queue* subsystem. The iPRP specification indicates that the implementation run on Linux kernel 3.11 and with *iptables* 1.4.12.

NFQUEUE allows iPRP to intercept en-queued UDP packets which are directed to the control and data ports of iPRP as well as the monitored application port by iPRP. Consequently, iPRP can easily modify the en-queued packets before they are sent to a lower network layer or forwarded to an application. After establishing an iPRP session, NFQUEUE rules on both communicating ends are configured to listen for packets that have session-relevant address information. The iPRP is located in the userspace and any application packets sent to monitored ports of active receivers are forwarded to iPRP. The payload of the forwarded packets is copied into new UDP packets. In these new packets (replicates), an iPRP header is prepended which contains the addressing information of the original packet. For each replicate, one of the IND-matched destination interfaces and the iPRP data port are used as the new destination information in the UDP/IP header. In this context, the number of replicates is the same as the number of matched receiver interfaces (IND matching). The original packet is dropped from the queue while the new iPRP data packets with the replicated payload from the original packet are submitted for transmission. On the receiver side, the NFQUEUE rule created by the IRD intercepts incoming packets for an iPRP session. These packets are processed in equivalent way as in the sender side of iPRP, but in reverse order. Here a blank packet is first created once one of the replicates arrives. This packet payload is filled with the acquired payload from the iPRP packet. Then, the iPRP header information is used to reconstruct the original UDP packet. After that, the packet is forwarded to its application. The iPRP packet and the following duplicates are dropped. The whole process for applications is transparent. This makes iPRP highly compatible with upper layer applications.

11.4 iPRP-RC4CPS

As indicated in Chapter 8, the evaluation of the existing MP communication protocols shows that iPRP is the most suitable candidate to implement RC4CPS.

In this Section, the implementation of the functions of RC4CPS in iPRP is described. The main target here is to provide seamless integration of the RC4CPS monitoring and selection mechanisms into the structure of the existing iPRP implementation. For this purpose, it was aimed to add the new features of RC4CPS in a way that they are separate from the original functions of iPRP. This way, modifications and additions in the future can be easily applied. This allowed preserving the existing implementation structure with clear and traceable interaction between iPRP and RC4CPS functions.

As mentioned previously, the current iPRP implementation has four daemons where one or more iPRP functions blocks (described in Section 11.2.2) are represented. These different tasks within the daemons are carried out using various routines that run as threads. Therefore, it was decided to implement RC4CPS as an own daemon. The daemon is called the iPRP Path-selection daemon (IPD). In fact, the use of daemons to separate the function blocks of iPRP provided a suitable interface to integrate the IPD. This is because the daemons mostly act in an autonomous way and interact with each other only by accessing and modifying the shared data structures. This in turn supported also the decision of implementing the functions of RC4CPS as an IPD instead of integrating those using additional routines inside the original daemons.

For implementing the IPD, it was required to adapt some of the data structures (indicated in Section 11.3.2) which store session-relevant information. This was necessary to support the change of the existing path matching mechanisms. In addition, a redesign for the initialization of the sender sockets was needed to enable their dynamic reconfiguration for the MP reselection. Moreover, some elementary interprocess communication (IPC) mechanisms were used to provide easy signaling and data exchange between the daemons.

11.4.1 iPRP limitations

In order to implement RC4CPS, it is necessary to modify the interface-matching mechanism of iPRP which exhibits multiple limitations.

1. *No full-mesh path configuration:*

The path matching of iPRP establishes e2e paths according to the connected sub-clouds and INDs of the available interfaces between the end-systems. In this context, the use of e2e paths that share one source- or destination interface is not allowed. More specifically, each path must have unique source and destination interfaces which are not used by any other path configuration. In the case of logically separated networks (sub-clouds), iPRP rely on the routing rules to ensure the logical separation of the utilized sub-clouds. Otherwise, the pursued idea of disjoint paths to increase redundancy would

be void and be not fulfilled. Unfortunately, the requirement of disjoint e2e paths cannot be guaranteed in the Internet compared to the case of dedicated WAN networks with controllable infrastructure. By contrast, RC4CPS approach takes into account the uncontrolled nature of the Internet.

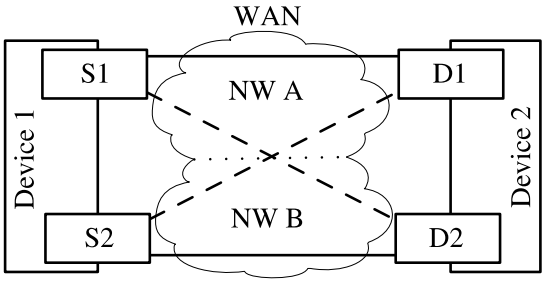


Figure 11.5 Example of two multihomed devices with interconnected networks A and B.

In RC4CPS, the requirement is to use path subsets with at least two source and two destination interfaces (5.14). However, subsets where two or more paths share the same interface but still utilize two interfaces at each end-system might be used. Such subsets should be avoided when there are other subsets where the paths provide higher redundancy (when they are all disjoint for example) and availability. If the setup in Figure 11.5 with two devices each with two network interfaces is considered, then in a typical iPRP setup and based on the network configuration, either the subset {(*S1*,*D1*), (*S2*,*D2*)} or {(*S1*,*D2*), (*S2*,*D1*) } can be used to duplicate data. If the sub-clouds *NW A* and *NW B* are logically separated, then only the path subset {(*S1*,*D1*), (*S2*,*D2*)} can be utilized. In RC4CPS, it is possible to use any of the two subsets as well as any of the 3-path and 4-path subsets as long as they fulfill the conditions in (5.14). In the case that there is no 2-path subset that fulfills the unavailability threshold in (5.14), there are still multiple 3-path subsets that can be used (e.g. {(*S1*,*D1*), (*S2*,*D2*), (*S1*,*D2*)}). An additional advantage of the RC4CPS MP selection is the possibility to use path subsets with higher number of paths compared to those of iPRP. More specifically, iPRP would fail when both paths (*S1*,*D1*) and (*S2*,*D2*) become unavailable. In the case of RC4CPS, the communication might continue over (*S1*,*D2*) when the subset {(*S1*,*D1*), (*S2*,*D2*), (*S1*,*D2*)} is selected (assuming that the failure is network related and not caused by the interfaces). From the above, it is necessary to allow each source interface to be matched with every destination interface to enable the path configuration of RC4CPS.

To address this issue, the original IND-matching was completely removed in iPRP-RC4CPS such that RC4CPS can consider all path subsets between two end-systems in the Internet that adhere to (5.14). In iPRP-RC4CPS, the concept of INDs is used only to retain most of the data structures as well as the functions handling them. This reduced the adaptation efforts as the INDs are used by the different function in iPRP to refer to the interfaces and e2e paths. Similar to the path representation of RC4CPS, a path in iPRP-RC4CPS is represented by a pair of INDs (*i*,*j*) where *i* is the IND of the source interface while *j* is the IND of the destination interface. In other words, the INDs in

iPRP-RC4CPS are used to refer to interfaces rather than the networks to which they are connected.

2. *No dynamic path reconfiguration:*

In iPRP, it is not expected to reconfigure the paths during an active session. They are configured only once after iPRP is launched and the interfaces' INDs are assigned. This is because, the assigned INDs to the interfaces control which paths will be established and used (see Section 11.2.1) unless the underlying network is reconfigured. RC4CPS, on the other hand, heavily depends on path reconfiguration. This is due to the fact that the availability of e2e paths is not guaranteed in the Internet environment. To ensure high reliability, the communication needs to be flexible in a way that different subsets of paths can be utilized to ensure the required availability of the communication service. With this regard, RC4CPS monitors the availability of the subsets of paths between communicating parties and reselects (periodically/urgently as described in Section 5.4) two subsets which adhere to (5.14) and have the minimum number of paths using (5.13). Therefore a dynamic reconfiguration of used paths for data transmission or even for monitoring is needed.

As a result, the ISD was modified to enable the reconfiguration of the sending sockets. This was achieved by redesigning the *Cache Routine* (Section 11.3.1) into the *Config Routine* (Section 11.4.2). Furthermore IPC had to be laid down to provide a flexible information exchange between all daemons. This is needed to communicate path changes to the ISD. The ISD then uses the new *Configuration Routine* to push the changes toward to the sending sockets.

11.4.2 Overview of modifications to iPRP

The main modifications carried out to integrate the RC4CPS approach with iPRP are best described by Figure 11.6 using the block diagrams of RC4CPS, iPRP, and iPRP-RC4CPS. The IND matching component at the iPRP sender was replaced by the MP Selection component of RC4CPS to provide the selected paths for data replication. In addition, the IND Advertiser at the iPRP receiver was removed in iPRP-RC4CPS. Moreover, the M&E components of RC4CPS were imported to the iPRP-RC4CPS architecture.

With regard to the original iPRP implementation, the major modification done to incorporate RC4CPS is the addition of the IPD daemon. This is illustrated in Figure 11.7 along with the major changes to the other daemons. In addition, there were several minor changes done in the different daemons. In this section, the applied changes to the individual iPRP daemons in order to support the addition of the IPD daemon will be detailed (the names of the modified/added C functions are also included). In addition, Appendix D lists the source files of iPRP-RC4CPS implementation and their functions.

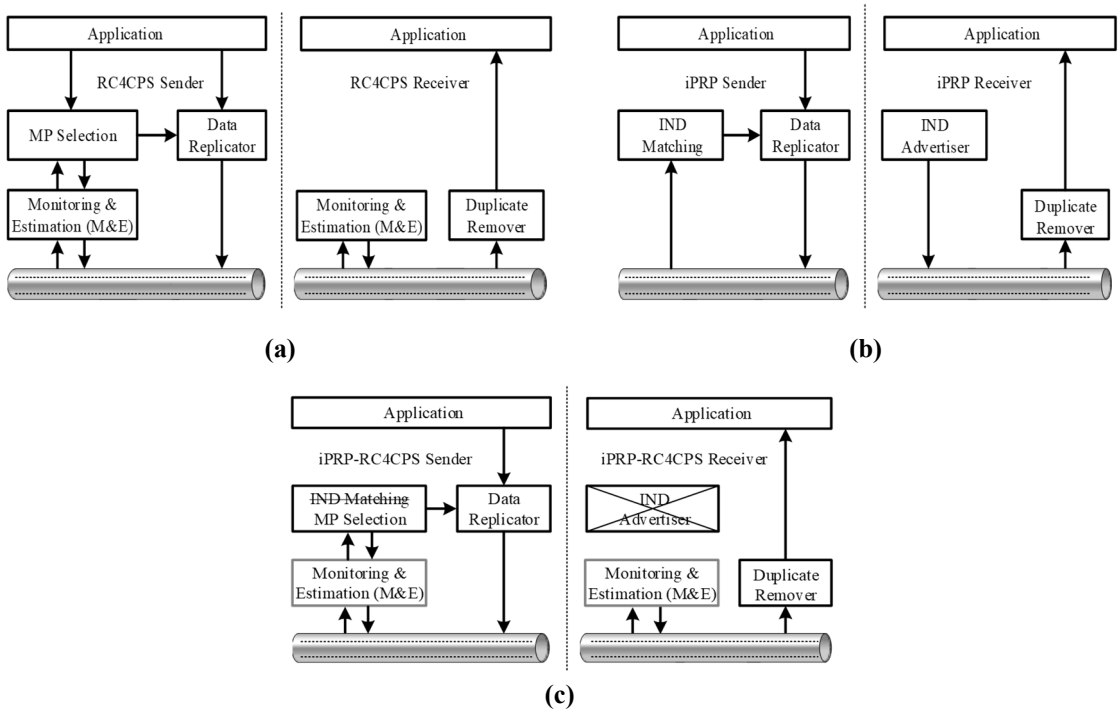


Figure 11.6 The integration of RC4CPS approach with iPRP: (a) RC4CPS architecture, (b) architecture of iPRP, and (c) the iPRP-RC4CPS architecture with the main modifications indicated.

1. *iPRP control daemon (ICD)*

Main function (*main()*):

- To allow communication between ISD and IPD, an initialization of pipes was added.
- To enable file-based configuration of interfaces and INDs, a function called *get_interfaces()* was added. This function along with the file-based configuration allows the dynamic reconfiguration of the selected paths by the MP Selection in the IPD daemon. The previous configuration of interfaces was done by providing the IP addresses of the interfaces to the ICD as command line arguments when lunching iPRP.
- To enable monitoring on the available interfaces on the receiver side and configuration of the corresponding sockets, the *host_store()* function was added. The function writes the information of all available host interfaces into a file. The file is used later to acquire the information about the available interfaces and to create the sockets to be used by the *pingreceive_routine()*. The information acquisition is done by the IPD using the *host_load()* function.

Receiver-ports routine (*receiver_ports_routine()*):

- The string arrays that contains the arguments to be provided to the IRD, IMD and IPD when launching using the new *launch_daemon()* function were

prepared. The function was added to simplify the creation of daemons. More specifically, the file descriptor of a pipe is passed to the new daemon using the *launch_daemon()* function. In addition, the unused in/out-ends of a pipe are closed in the parent and child processes dynamically by the same function.

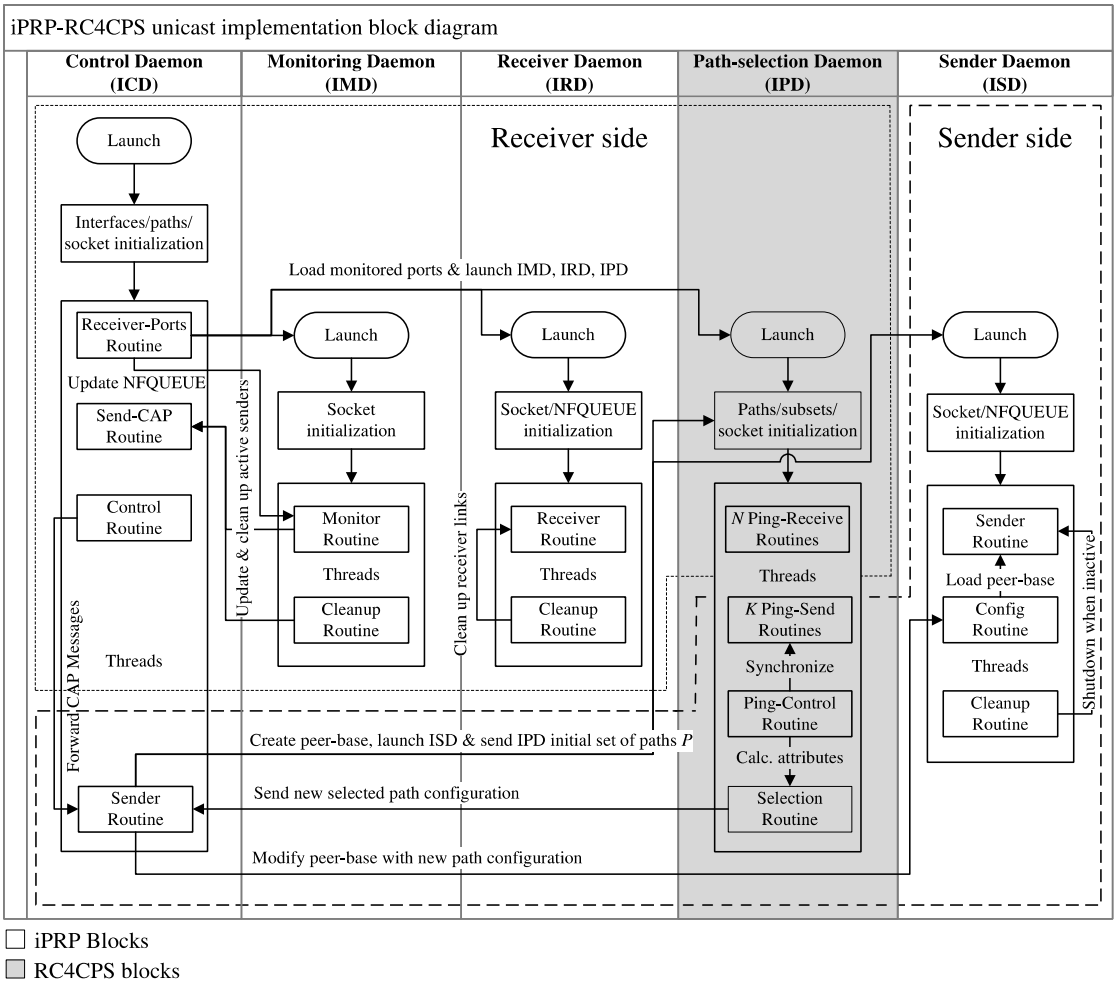


Figure 11.7 Architecture of the iPRP-RC4CPS implementation including the IPD daemon and the major modifications to the other daemons to support it.

Sender routine (*sender_routine()*):

- The *ind_match()* function was removed as it is not utilized in the iPRP-RC4CPS anymore (see Section 11.4.1)
- The ISD is launched using the new *launch_daemon()* function.
- A pipe between the ICD and IPD daemons was created to instruct the launch of the IPD after an *iPRP_CAP* message is received. This also launches the sender-side routines of the IPD for the monitoring and selection as it is aware that the machine participates now in the iPRP-RC4CPS session as a sender.

- The function *get_selectedpaths()* was added for the initial file-based path configuration. This is needed after lunching the ISD using the *sender_routine()* for the first time. In this case, the path configuration is read from the specified file and provided for packets duplication until the selected subsets of paths are provided by the IPD.
- The *peerbase_insert()* function was heavily modified to make the creation of peer-bases using the new path-selection mechanisms. In iPRP, the INDs of the receiver were first compared with the sender ones. Then, matching pairs were identified. The *get_iface_from_ind()* function is used after that to inquire about the corresponding interfaces of the matching INDs and to create a path entry for each in the peer-base. By contrast, the selected paths in iPRP-RC4CPS are passed directly using the new selected paths array obtained from the IPD. Only at the start of iPRP-RC4CPS, the array for the new selected paths is populated with the paths acquired using the function *get_selected_paths()* from a configuration file.
- The ICD-IPD pipe is polled after each reselection in the IPD to communicate the path configuration changes.
- The function *return_sender_links()* was added to obtain all current sender links. This is needed to reconfigure all active peer-bases in accordance to the newly selected paths through the *peerbase_insert()* function.
- The ICD-ISD pipes are utilized after paths reselections done by the IPD to broadcast modifications in the path configuration. The ISD then reconfigures its sending sockets correspondingly.

2. *iPRP sender daemon (ISD)*

Main function (*main()*):

- A pipe initialization was added to communicate with the ISD. To achieve this, the ICD passes the file descriptor for the pipe when launching the ISD where it is assigned to a variable.
- The *cache_routine()* was replaced with a new *config_routine()*.
- One additional 1-byte IND field was added to the original iPRP header in order to support the RC4CPS representation of e2e paths which utilizes both the source IND and the destination IND.

Configuration routine (*config_routine()*):

- The *cache_routine()* was replaced by the *config_routine()*. The routine takes care of loading the peer-base and configuring the required send sockets used by the *send_routine()* based on the initial path selection. After that, the routine polls

the ICD-ISD pipe which signals paths reselection. If a reselection is done, then the corresponding peer-base is configured by the ICD with the new paths. Then, the *config_routine()* reloads the peer-base as well as removing the old send sockets to create new ones for the new path selection.

Sender routine (*send_routine()*):

- A conditioning variable was added to avoid send socket reconfiguration by the *config_routine()* while data are being duplicated.

11.5 Path-selection daemon

11.5.1 Path monitoring:

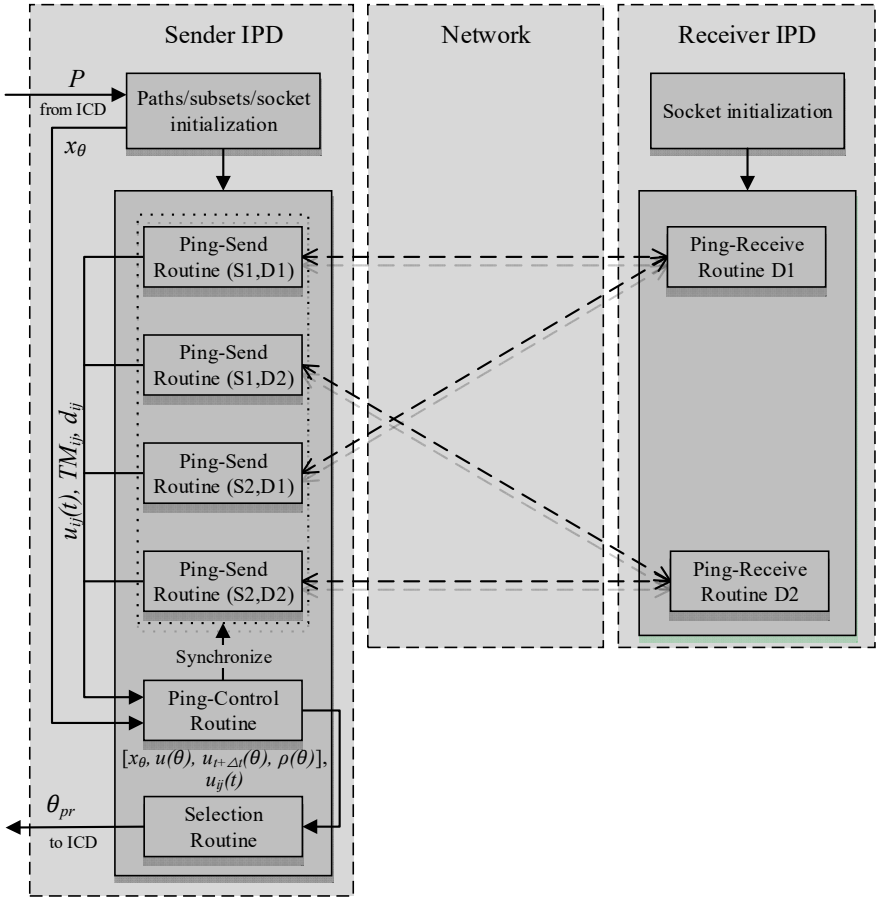


Figure 11.8 IPD's Block diagram assuming the MP setup in Figure 11.5.

As shown in Figure 11.7, the structure and initialization of the IPD daemon follows that from the other iPRP daemons. The daemon is launched with the IMD and IRD of iPRP-RC4CPS and has its own threads to manage and to launch. A more detailed block diagram of the IPD is shown Figure 11.8. The e2e paths monitoring is carried out by the *Ping-Receive Routines* (PRRs) (Alg. C-1) as well as the *Ping-Send Routines* (PSRs) (Alg. C-2). On IPD start, N PRRs are launched, where N is the amount of available

network interfaces. Each PRR creates and configures a receive socket and polls it on the iPRP-RC4CPS ping port for incoming probe packets. Once a probe packet arrives at a PRR at the receiver, the sender network address is reused as the new destination address and the same packet is returned as an ACK.

When an iPRP-capable device participates in a session as a sender, the ICD starts the ISD for data replication and provides the initial set of available paths (P) to the *main()* function in the IPD. P is acquired by the *Sender Routine* of the ICD by reading the configuration from a file using the *get_selected_paths()* function. The IPD proceeds to launch the *Ping-Control Routine* (PCR) (described by Alg. C-3), the *Selection Routine* (SR) (described by Alg. C-10), and N PSRs, with N being the size of P. For example, if both the sender and receiver have two interfaces, then $P = \{(s1,d1),(s1,d2),(s2,d1),(s2,d2)\}$ and $|P|= 4$. Before starting the monitoring, the PSRs configure their send sockets using the *pingsend_setup()* function and hold on till the PCR starts the monitoring. Once the monitoring started, the send of probe messages on all paths is triggered by the PCR every T_{PING}. Each PSR then sends a probe message over the corresponding e2e path and waits for its ACK. The probe messages in the iPRP-RC4CPS implementation are simple UDP packets with an 8-byte integer in the payload that carry the probe sequence number (PSN). This PSN is incremented with every iteration triggered by the PCR. Delayed packets from previous probes received while the PSR is waiting for the ACK of the current probe packet are discarded. A path is considered as unavailable ($u_{ij}(t) = 1$) if no ACK is received within $T_{TIMEOUT}$. In this case, the unavailability probability for the next transmission, p_{11} (provided in (5.8) and assuming the Gilbert model), is calculated. When a probe is acknowledged, then the RTT of the probe packet over the path (d_{ij}) is logged. In addition, the probability p_{01} as given in (5.8) that the path becomes unavailable for the next transmission is calculated. The path also is declared as available ($u_{ij}(t) = 0$) if it was unavailable during the last transmission. After the transmission and processing of a probe, the PSR signals that it is ready to the PCR. The PCR triggers another transmission only after all PSRs are finished.

11.5.2 Attributes Calculation

In the iPRP-RC4CPS implementation, the PSRs already perform a partial acquisition of the attributes for MP selection. More specifically, the attributes which can be calculated without having knowledge about the other path's behavior, is done in the PSRs. These include $u_{ij}(t)$, $u^*_{ij}(t+\Delta t)$), and the last value in d_{ij} of the monitored path. By contrast, the PCR calculates $\rho(\theta)$ as it requires information from multiple PSRs. The PCR also combines the individual path statistics for the different subsets and updates their profiles accordingly. It is important to indicate here that only the Gilbert model was considered and implemented for unavailability prediction in iPRP-RC4CPS.

The power set $\mathcal{P}(P)$ is calculated at the IPD after receiving P from the ICD. This is done by using the *pathPowerset()* function which is part of the *main()* function. Only 2- and 3-path subsets from the $\mathcal{P}(P)$ were considered in this implementation, provided that the RC4CPS selection criteria in Section 5.2.3 were fulfilled. According to these conditions, subsets that use only one source or one destination interface are not considered. 4-path subsets were not considered to reduce the computation efforts during runtime where the measurements conducted in Chapter 6 indicated that the availability for 3-path subsets was always 100% (given that the conditions in (5.14) are satisfied). The attributes of the considered subsets x_θ including $u(\theta)$, $u_{t+\Delta t}(\theta)$, and $\rho(\theta)$ are continuously updated by the PCR in every iteration. This is done using the $u_{ij}(t)$, TM_{ij}, and d_{ij} (if the path was available) values provided by the PSRs. It is necessary to mention here that the sample Pearson's correlation coefficient $r(d_{ij}, d_{mn})$ given as:

$$r(d_{ij}, d_{mn}) = \frac{\sum (d_{ij} \bullet d_{mn}) - \frac{\left(\sum d_{ij}\right)\left(\sum d_{mn}\right)}{s}}{\sqrt{\left(\sum \left(d_{ij}\right)^2 - \frac{\sum \left(d_{ij}\right)^2}{s}\right)\left(\sum \left(d_{mn}\right)^2 - \frac{\sum \left(d_{mn}\right)^2}{s}\right)}} \tag{11.1}$$

was used in the implementation to compute equation (5.6). This allowed the calculation of correlation in real time and on sample bases.

The correlation can be computed only if the paths are available as it is done based on the RTTs. Otherwise, the correlation calculation is not feasible or might lead to erroneous results, especially when 3-path subsets are considered. In this case, the datasets of time delays would have different sizes and their correlation values would be not comparable. Therefore, the correlation calculation was skipped globally for all subsets if one or more paths experienced an outage during the current transmission. In this case, the last valid correlation coefficient is used where all the considered paths were available.

11.5.3 MP Selection

The *Selection Routine* (SR) manages the path selection where it performs the periodic as well as the urgent reselection (described in Alg. C-10). The routine can be considered to be clocked by the PCR and, therefore, performs the inspection of θ_{pr} and θ_{ba} after the subsets attributes are updated by the PCR. θ_{pr} and θ_{ba} are exchanged either when θ_{pr} has unavailability higher than u_r or has a higher unavailability probability $u_{t+\Delta t}(\theta)$ compared to that of θ_{ba} for a certain time t_e. Then, the new θ_{pr} is provided to the ICD using the *subset_reconfiguration()*. Exchanging θ_{pr} and θ_{ba} due to higher unavailability probability of θ_{pr} is done after some time t_e as described in Section 5.4. The aim of such exchange delay is to avoid triggering an exchange due to a single unavailability event on one path only. For the periodic reselection, the SR uses a reselection counter that is incremented after each iteration triggered by the PCR. In the current implementation, it is executed in the 30th iteration (30th T_{PING}). When there is an urgent replacement which is triggered when, for example $u(\theta)$ of θ_{pr} and θ_{ba} exceeds u_r, then the reselection

counter is increased. This is done to speed up the periodic reselection and to find a new θ_{ba} sooner. The reselection of θ_{pr} and θ_{ba} is done using *subsetselection()* function (described in Alg. C-11) and according to (5.13) and (5.14).

11.5.4 MP Reconfiguration

After an urgent or periodic reselection, the selected θ_{pr} is provided to the ICD. The ICD uses new paths in θ_{pr} to update the peer-base and signals the changes of the peer-base using a pipe to the ISD. The ISD immediately proceeds to reloading the peer-base and to do a socket reconfiguration through its *Configuration Routine*. Condition variables were provided to avoid data replication while socket reconfiguration is being applied. Once the reconfiguration is done, the next data packet to be sent by iPRP-RC4CPS is replicated on the new selected paths provided by θ_{pr}.

11.6 Evaluation

This section focuses on three aspects. First, the degree of redundancy that iPRP-RC4CPS can provide compared to iPRP is discussed. Then, the communication overhead using the dynamic MP selection of iPRP-RC4CPS is compared to that of iPRP. Second, the diversity and unavailability probability estimations are evaluated by impacting the individual paths/subsets. Lastly, the achieved availability by iPRP-RC4CPS is compared to the required availability by high demanding CPS applications like smart grids (Section 11.6.3). As indicated in Section 3.4 and Chapter 8, almost all existing MP protocols are either throughput-oriented, were proposed for dedicated networks, or do not support path selection. Therefore, it was not feasible to compare iPRP-RC4CPS with existing MP protocols such as MPTCP.

As iPRP-RC4CPS targets increasing availability, factors such as delay and throughput were not considered in details in this section. Nevertheless, the maximum tolerant delay of the application using iPRP-RC4CPS can be imposed as the timeout of the monitoring packets. This will allow marking individual paths as unavailable when the time delay experienced by the monitoring packets is higher than maximum tolerant delay of the application.

The PlanetLab and NorNet platforms were not utilized to evaluate iPRP-RC4CPS in real-world due to the same issues described in Sections 6.1 and 10.2.

11.6.1 Redundancy and Overhead of iPRP and iPRP-RC4CPS

1. *Redundancy Increase*:

iPRP requires pre-configured networks with logically or physically separated paths. In addition, the path setup in iPRP is static and only one path subset can be used during a session. Therefore, the increase in redundancy when using iPRP with such path configuration for dedicated networks is limited. In iPRP-RC4CPS, however, the

dynamic path reconfiguration combined with the diversity estimation provides a higher level of redundancy. If the setup in Figure 11.5 is assumed for iPRP with $n = k = 2$, only the subsets $\{(1,1),(2,2)\}$ and $\{(1,2),(2,1)\}$ can be utilized. If the chosen subset becomes unavailable, then the communication session fails because switching to the other subset is not possible. For $n = k$, the iPRP path redundancy is $1 + (k - 1)$ with only one subset to select. With iPRP-RC4CPS, all available paths can be used concurrently and the currently used subset of paths can be changed when it starts to experience degradation. For $n = k$, the iPRP-RC4CPS path redundancy is $1 + (k^2 - 1)$ with additional subsets to choose from given by:

$$\sum_{l=2}^{k^2} \binom{k^2}{l}, \tag{11.2}$$

where $k^2 = P$ in this case. If the conditions in (5.14) are considered, then iPRP-RC4CPS does not consider all these replacements where some use only one source/destination interface. For $n = k$, the number of these unconsidered subsets is given by:

$$\sum_{l=1}^{k-1} 2kT_l, \tag{11.3}$$

where T_l is a triangular number that counts the objects that can form an equilateral triangle with l dots on a side such that $T_1 = 1$, $T_2 = 3$, $T_3 = 6$, etc. Equation (11.3) represents the number of subsets out of those in (11.2) that are not used by iPRP-RC4CPS.

2. *Communication Overhead Reduction*:

As mentioned previously, RC4CPS targets reducing the utilization of network resources while maintaining a certain level of availability. If the overhead of iPRP and that of iPRP-RC4CPS are compared using Figure 11.9, then iPRP would continue using the same subset of three paths $\{(s1,d1),(s2,d2),(s3,d3)\}$ even if two paths are already providing adequate availability or when two paths start to experience frequent unavailability events. iPRP-RC4CPS on the other hand would use two paths only and, when necessary, reselect another subset of two paths. If *Device 1* in Figure 11.9 is a PMU in a smart grid that samples the current and voltage values 50 times per second, then an overhead of $3 \cdot 50 \cdot 98$ bytes = 14700 bytes is generated where 98 bytes is the size of an iPRP packet without payload. If iPRP-RC4CPS is monitoring all e2e paths every second, then the monitoring overhead is $2 \cdot 9 \cdot 64$ bytes = 1152 bytes where 64 bytes is the size of an iPRP-RC4CPS monitoring packet that is sent first from sender to receiver and then back from receiver to sender. In addition, the data overhead of iPRP-RC4CPS using two paths is $2 \cdot 50 \cdot 99$ bytes = 9900 bytes. Hence, the iPRP-RC4CPS total overhead is 1152 bytes + 9900 bytes = 11052 bytes (24.8% reduction compared to the iPRP case). It is necessary to indicate that both iPRP and iPRP-RC4CPS use the capability messages which add to the overhead (but are not considered in this calculation).

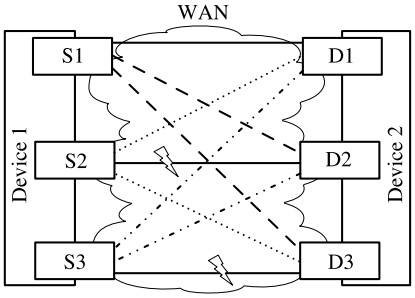


Figure 11.9 Exemplary MP setup between two devices, each with 3 interfaces.

11.6.2 Measurements in Lab Environment

1. *Evaluation Setup*

This first setup was built in a lab to evaluate iPRP-RC4CPS in a controlled environment and ensure that it works as desired. As illustrated in Figure 11.10, the communicating end-systems used in the setup are two PCs where both run Ubuntu 16.04 OS with kernel versions 4.4.0-59 and utilize *iptables* 1.6.0. On each PC, two dedicated network cards are available. To create a network environment that is similar to the Internet, a network emulator namely the PacketStorm1800E [153] was utilized. More specifically, the behavior of the Internet is emulated by impairing the data and probe packets of iPRP-RC4CPS using delays and packet drops with random distributions.

As Figure 11.10 shows, each network interface of the two PCs is in a different network. Due to a limited number of Ethernet ports in the PacketStorm and the routers, physically disjoint e2e paths could not be realized. The connections between *Router 1* to *Router 3* and from *Router 2* to *Router 4* are realized through their serial interfaces. The use of the innermost routers (*Router 3* and *4*) was mainly done to connect the PacketStorm between *Router 1* and *Router 2*.

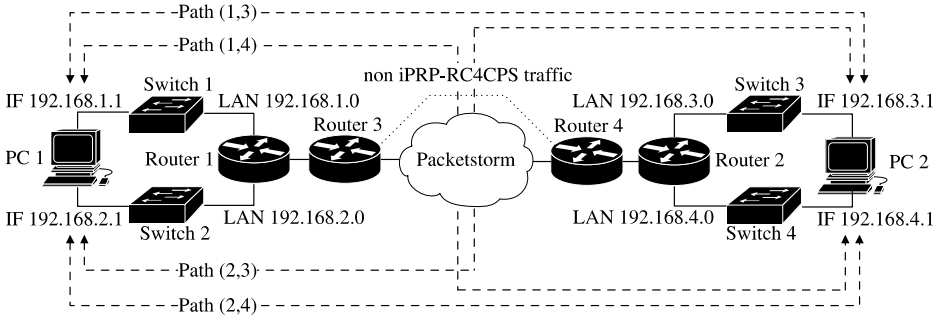


Figure 11.10 iPRP-RC4CPS evaluation setup in lab environment.

The logical e2e paths between the PCs are also illustrated in the figure, namely: (*1*,*3*), (*1*,*4*), (*2*,*3*) and (*2*,*4*). With the limited resources, this arrangement was sufficient to evaluate iPRP-RC4CPS and its mechanisms. It is assumed in the above setup that the shared routers outside the environment of the PacketStorm do not exist. To remove the effect of background traffic on the simulation, the innermost routers were configured

with policy-based routing such that only UDP streams of iPRP-RC4CPS (including the data, control and probe ports) pass to the PacketStorm.

2. *Correlation test*

To demonstrate the impact of the correlation calculation on the MP selection (Section 5.2.1), the configuration of the PacketStorm shown in Figure 11.11 was used. As shown in the figure, each e2e path is impacted using one or more blocks. Most of these blocks are time delay blocks with uniform delay distribution with minimum and average delay parameters as indicated in the figure. In addition, the paths (*1*,*4*) and (*2*,*3*) were impacted by a common, time-triggered delay impairment block of fixed 30ms. The delay was randomly triggered by a uniform distribution with the parameters depicted in the figure.

All the delay blocks with random delay are intended to simulate uncorrelated delays. The constant delay block that is shared between the e2e paths (*1*,*4*) and (*2*,*3*), which has a random duration but concurrent trigger, represents a congested shared link in the Internet. The values for the delays and trigger timer were chosen empirically. The delay values are relatively high. This is mainly to elevate the impact of the shared links between the routers due to hardware limitations mentioned previously, where the *Routers 1* and *3* and the *Routers 4* and *2* was interconnected using serial ports with low capacity. This resulted in correlated idle latencies of 15-45 ms. Therefore, it was necessary to configure the impairment blocks such that the impact the network setup exhibited on the measured delays could be compensated.

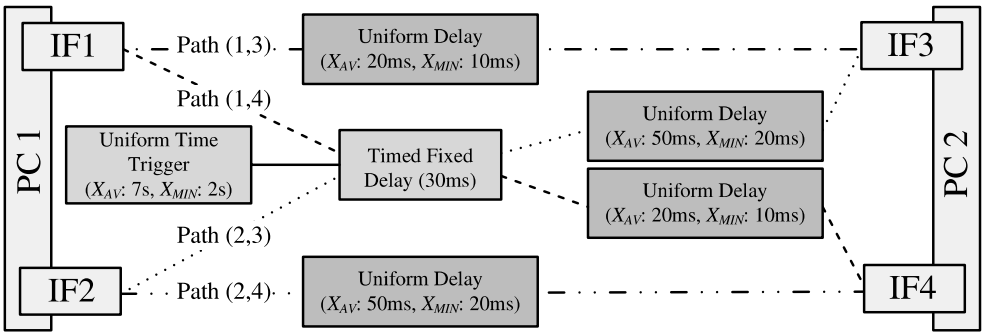


Figure 11.11 The PacketStorm configuration for the correlation test where X_{AV} and X_{MIN} are the average and minimum time delay of the uniform delay distribution.

For the first 30 minutes, the setup in Figure 11.11 was run without enabling the shared timed delay. Figure 11.12 shows the correlation values for the considered 2- and 3-path subsets using (11.1) over the test interval. The plot in Figure 11.11 as well as the following ones is direct representation regarding the data available for iPRP-RC4CPS to perform the MP selection during runtime. Here T_{PING} was set to 1 s and the periodic reselection interval to 30 s. Within 15 minutes from the test start, the values largely fluctuated before they started to converge towards more stable values. This is attributed to the low number of samples at the beginning of the test interval.

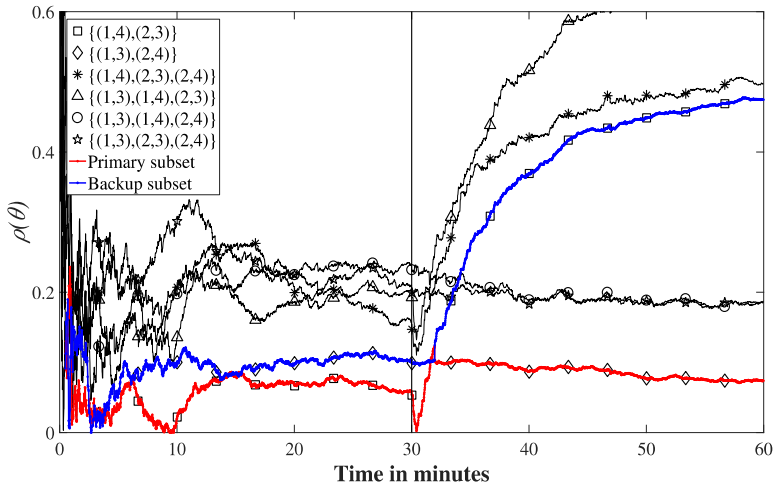


Figure 11.12 Correlation values of all monitored subsets and the selected θ_{pr} and θ_{ba} subsets (lab setup).

After 30 minutes, the congestion model on the paths (*1*,*4*) and (*2*,*3*) was started using the fixed timed delay block. The block was enabled and disabled in a random manner and within 2 to 12 second intervals. Before the congestion model was activated, {(*1*,*4*), (*2,3*)} and {(*1*,*3*),(*2*,*4*)} exhibited a comparable degree of correlation which is attributed to the their identical block configuration within the environment of the PacketStorm. After starting the congestion, the $\rho(\theta)$ value for the subset {(*1*,*4*), (*2,3*)} as well as all 3-path subsets that include the same 2-path subset directly started to rise. Even though the subset {(*1*,*4*),(*2,3*)} was selected as θ_{pr} prior the start of congestion, iPRP-RC4CPS quickly changes its selection for θ_{pr} to {(*1*,*3*),(*2*,*4*)} through periodic reselection. Due to the lack of unavailability events, the selection in this simulation setup was only based on the correlation (as $u_{t+\Delta t}(\theta)$ was 0). This also clarify why the system selected a highly correlated subset (i.e. {(*1*,*4*), (*2,3*)}) for θ_{ba} after the congestion model was started. According to the RC4CPS selection criteria in Section 5.4, backup subsets need to have different paths from the primary subset. Although {(1,3),(1,4),(2,4)} and {(1,3),(2,3),(2,4)} exhibited lower correlation values compared to the other subsets, these subsets were not considered since they inherited the e2e paths from θ_{pr} (have only one additional path). As a result, the MP Selection component in iPRP-RC4CPS chooses the subset {(*1*,*4*),(*2,3*)} for θ_{ba}.

This test showed that the system can detect a shared link and react upon it. Although there was no indication of jointness during the first 30 minutes between the paths, the correlated patterns of packet delays were detected shortly after activating the congestion model and θ_{pr} was chosen accordingly.

3. *Prediction test*

To test the impact of unavailability prediction on MP selection, the PacketStorm was configured as depicted in Figure 11.13. The e2e paths (*1*,*4*) and (*2*,*3*) run through drop impairment blocks. These blocks are configured to drop 5% of the traffic in 2 packet bursts fashion. The drop impairment block on path (*1*,*3*) drops only 3% but in a similar

2 packet bursts fashion. By contrast, the drop impairment for the path (*2*,*4*) was not activated for the first 30 minutes. After its activation, the impairment block causes 20% burst drop (after the first 30 minutes). This setup aims to evaluate the system's reaction if a subset, such as {(*1*,*3*),(*2*,*4*)} which has low unavailability and is selected for θ_{pr}, suddenly started to experience degradation. For clarity of presentation, the attributes of the 3-path subsets were almost not considered in this section.

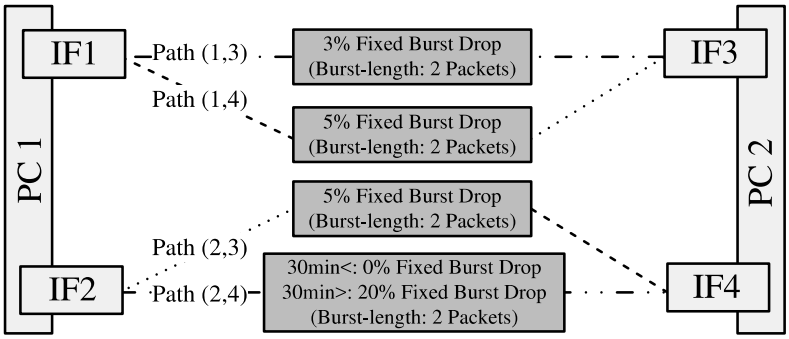


Figure 11.13 The configuration of the PacketStorm for simulating e2e paths with random and bursty packet drops.

In Figure 11.14c and Figure 11.14d, $u_{t+\Delta t}(\theta)$ of both subsets of paths {(*1*,*4*),(*2,3*)} and {(*1*,*3*),(*2*,*4*)} is shown over time. Because $u_{t+\Delta t}(\theta)$ provides the probability of future unavailability based on the current state, it usually fluctuates between two levels that are corresponding to the two probabilities p_{01} or p_{11} that are estimated based on status of the current probe (Section 5.2.2). More specifically, $u_{t+\Delta t}(\theta)$ returns p_{01} when the last probe is successful and p_{11} otherwise. The p_{01} values can be seen at the bottom of the figures. Every time there is a loss of a probe packet, the state in the Gilbert model changes to 1 and $u_{t+\Delta t}(\theta)$ returns p_{11} instead. These p_{11} values are the peaks in the figures. The probability p_{01} of switching from a loss-less state to a loss-state is generally very low and might decrease over time if no packets are lost. By contrast, the peaks in the figures show the values for p_{11} and indicate unavailability events on one of the paths of the subset. Values close to 1 indicate concurrent losses on both paths, as the probabilities are summed in $u_{t+\Delta t}(\theta)$ according to (5.9). Concurrent losses lead to an increase of the average unavailability, $u(\theta)$, visible in Figure 11.14b. For this test, the unavailability constraint u_r was set to 0.01 to not interfere with the selection process and let the system base its subset choice only on the unavailability prediction.

Figure 11.14c clearly shows how the unavailability events on subset {(*1*,*4*),(*2,3*)} are evenly distributed over the whole sample time because the paths were only subject to the constant 5% burst drop. In Figure 11.14d however, the increased drop rate on path (*2*,*4*) caused a massive increase in single and concurrent unavailability events. It also shows how p_{01} values (of the paths subset, and more specifically of the impacted path (2,*4*)) started to rise after the quantity and frequency of burst losses increased.

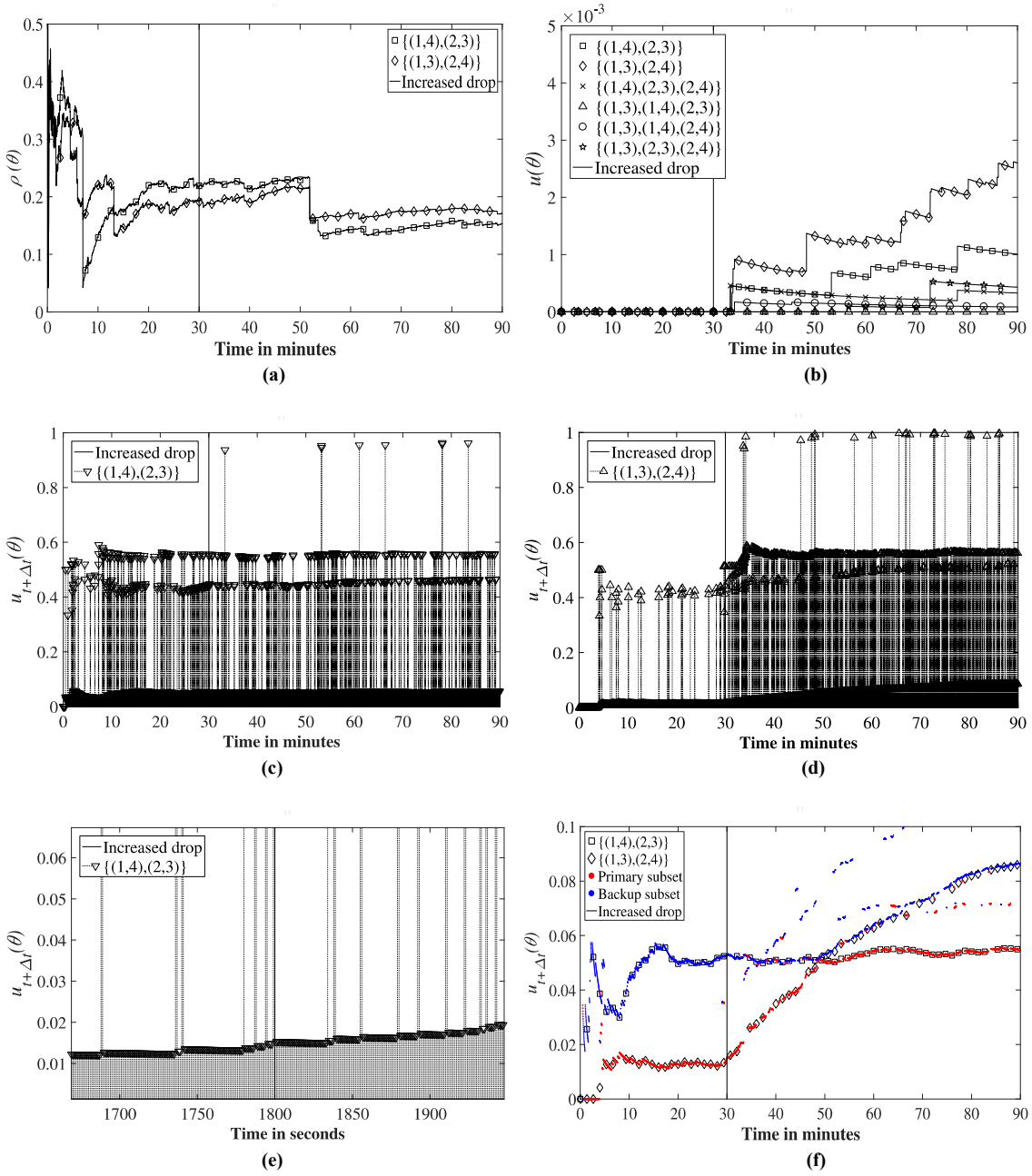


Figure 11.14 Results from the prediction test in the lab environment: (a) correlation between the 2-path subsets, (b) unavailability of all subsets of paths, (c) unavailability probability of subset {(*1*,*4*),(*2*,*3*)}, (d) unavailability probability of subset {(*1*,*3*),(*2*,*4*)}, (e) detailed view of Figure 11.14c highlighting the impact of the increased drop on the course of p_{01}, (f) the course of only p_{01} for the 2-path subsets and the selected θ_{pr} and θ_{ba} (red for θ_{pr} and blue for θ_{ba}).

Figure 11.14e gives a more detailed view of the individual samples and shows how p_{01} values progressively increases. Whenever there is a burst of packet losses, indicated by the higher stems reaching out of the axis limits and representing p_{11}, the values are slightly shifted upwards (as p_{01} given in (7.2) increases as the number of unavailability bursts increases). Over a longer period, frequent packet losses lead to an increase of $u_{t+\Delta t}(\theta)$ (as p_{01} and p_{11} values increase too). The bursts in this example consist of 2

packet drops as configured in Figure 11.13. The isolated single drops, visible in the diagram, were identified to be happening in the traces from the network interface *IF2* and were not intended. However, they did not affect the gained conclusions of this test. The diagrams show how unavailability events on a single path can be identified by the MP selection through p_{11}. Moreover, the frequency and severity of bursty packet losses have a direct impact on p_{01} and can be used to quantify the subsets according to the probability of future losses.

In Figure 11.14f the course of p_{01} for both 2-path subsets is highlighted. Also, the selection of the primary and backup subsets is depicted. The figure shows how the MP selection preferred the subset with the lower unavailability probability. Initially the subset {(*1*,*4*),(*2,3*)} was avoided due to its bursty packet losses and because {(*1*,*3*),(*2*,*4*)} posed a better alternative. However, after path (*2*,*4*) experienced severe degradation in the form of increased packet drops, the system detected the contingency and switched to a subset without the degraded path. Due to the minimization mechanisms, the other 2-path subset was favorited over the 3-path subsets to be selected for θ_{ba}. The gaps in the figure regarding the selected subset are due to the limited range of the Y-axis. As a result, some of values are not within the considered range in the figure. These are mostly the values associated with p_{11} which is obtained during unavailability durations. t_e (Section 11.5.3) for exchanging θ_{pr} and θ_{ba} was set to 5. As mentioned previously, this targets avoiding rapid and unnecessary subset switching that might happen when short bursty drops occur on a single path. The dots in the figure that do not follow the course of any of the 2-path subsets belongs to the short time intervals where a 3-path subset is selected when neither one of the 2-path subsets fulfilled the selection criteria. Such situation occurs when the periodic reselection is started while the 2-path subsets have concurrent unavailability on all paths or have single-path unavailability and, consequently, high unavailability probability. Nevertheless, the 2-path subsets were usually recovering in a short time and the IPD reselected these subsets again within the next reselection interval.

As the RTTs of the probe packets over the different paths were not influenced in this setup, $\rho(\theta)$ was expected to exhibit similar values for both of the 2-path subsets and, consequently, will not be a decisive factor in the selection of θ_{pr} and θ_{ba}. Figure 11.14a indicates the correctness of this assumption. With this regard, a similar but high $\rho(\theta)$ values for both subsets can be observed. The high values for $\rho(\theta)$ are attributed to the previously highlighted drawbacks of the utilized hardware setup. The sudden drops in the curves around the middle of the test interval are due to abrupt and large deviations in the RTT delays from their average values. These changes in RTTs were unintended and caused at a random time by the network equipment. The inspection of the log files indicated that after 52 minutes from the start of the test, a single RTT value of 364 ms from interface *IF2* was logged while the average RTT values are around 15-45 ms. This, consequently, led to the abrupt decrease in $\rho(\theta)$ values for both subsets. When such high

RTT delay is experienced by packets on both paths of a subset, then $\rho(\theta)$ would increase rather than decrease. This observed impact of RTT values further supports the mechanism of using the path's cross correlation to estimate e2e paths diversity.

The carried out test proved that the proposed system is able to detect fluctuations in e2e paths unavailability and do the selection of θ_{pr} and θ_{ba} using the acquired information. Here subsets with 2 paths were compared during time intervals with increasing frequency of unavailability events. The results in this test show how iPRP-RC4CPS was selecting the subset with lower temporal unavailability throughout the test course regardless of the initial unavailability probability of the different subsets. This also assists the utilization of unavailability prediction to identify the temporal unavailability characteristics for the monitored subsets of paths and improve the MP selection to reflect the dynamic behavior of Internet paths.

11.6.3 Measurements in the Internet

1. *Evaluation Setup*

An ideal setup for testing iPRP-RC4CPS in the Internet is to have different ISPs along with static public IPs for each of the network interface. However, static IPs are usually offered for business class services with extra costs. In addition, the procedures for such services take more time to obtain such IPs and advertise them in the routing tables of routers belonging to the service providers. To address the first requirement of different ISPs, three cellular and one wired connections were utilized as shown in Figure 11.15. For the second requirement of static public IPs, a virtual private network (VPN) service running on a server in the Amazon network was utilized. This service allowed creating one virtual network for each interface and forwards the traffic between the virtual networks using routing rules in the Linux-based server. The use of a single server for the VPN service is to use the *Linux netem* (Network Emulation) traffic control facility [158]. This allowed impacting the different flows between the network interfaces by impairments such as time delay and packet loss to further evaluate iPRP-RC4CPS. Nevertheless, the use of iPRP-RC4CPS for a real world smart grid application would necessitate a different server for VPN service between each pair of interfaces between the source and destination to improve reliability and elevate the single point of failure created by using a single server.

Similar to the configuration from Section 11.6.2, the PCs in Figure 11.15 both run Ubuntu 16.04 with kernel version 4.4.0-59 and *iptables* 1.6.0. In addition, the network interface cards on each PC connect to two different ISPs. Figure 11.15 also illustrates the traversed ISPs by *Traceroute* probes to the server and the assigned number to each network interface. With this setup four logical e2e paths between *PC1* and *PC2* were created: (*1*,*3*), (*1*,*4*), (*2*,*3*) and (*2*,*4*). As each path was established over a VPN tunnel, the corresponding IPs used in iPRP-RC4CPS are those of the VPN tunnels. Moreover,

T_{PING} was set to 1 s and the periodic reselection interval was set to 30 s in iPRP-RC4CPS.

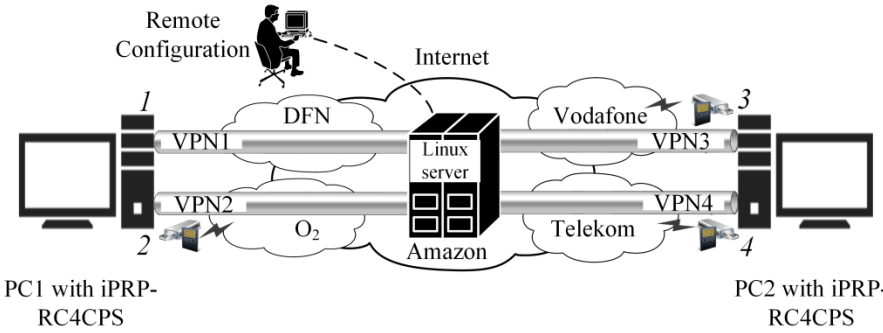


Figure 11.15 iPRP-RC4CPS evaluation setup in the Internet environment.

Before starting the analysis of the results, the plotting method of the results is shortly explained. As mentioned above, T_{PING} was selected to be 1 s. However, plotting all data samples might result in unclear figures especially in the case of the data representing $u_{t+\Delta t}(\theta)$ values. This is because $u_{t+\Delta t}(\theta)$ values might fluctuate between different levels based on the state of the path and the corresponding event probability (p_{01} and p_{11} in Section 5.2.2). Therefore markers plotted in intervals of 80 samples are used (except in the case of the markers for θ_{pr} and θ_{ba}) to avoid plotting them on top of each other in the following figures.

2. *Correlation test*

To demonstrate the impact of the correlation calculation on the MP selection, the *Linux* server was configured as follows: The duration of the test was 90 min to collect a large number of samples and yield better estimation. In the first half hour of the test, no impairments were applied. Then, a congestion model configured using the *Linux netem* on the paths (*1*,*3*) and (*2*,*4*) was started. The model was activated and deactivated for a random number of seconds in the intervals [1,10] and [1,60] respectively and in a sequential manner. When the model is active, the packets on (*1*,*3*) and (*2*,*4*) receive a random delay from a truncated normal distribution with a mean value of 20 ms, upper bound of 30 ms, and lower bound of 10 ms. The configuration of the congestion model is motivated by the results obtained in [154] regarding the maximum single hop delays in core IP networks.

Figure 11.16 shows the correlation values for the considered 2- and 3-path subsets using equation (5.6) in relation to the time. This and further plots are direct representations of the data that is available to the MP selection of iPRP-RC4CPS during runtime. For clarity of presentation, values higher than 0.2 are not illustrated. This is because the test targets illustrating how the MP selection is impacted by the value of $\rho(\theta)$ of the different subsets. Within the first 20 min, the values heavily fluctuate before starting to converge towards a steady level as more samples were gathered. After 30 minutes, the congestion

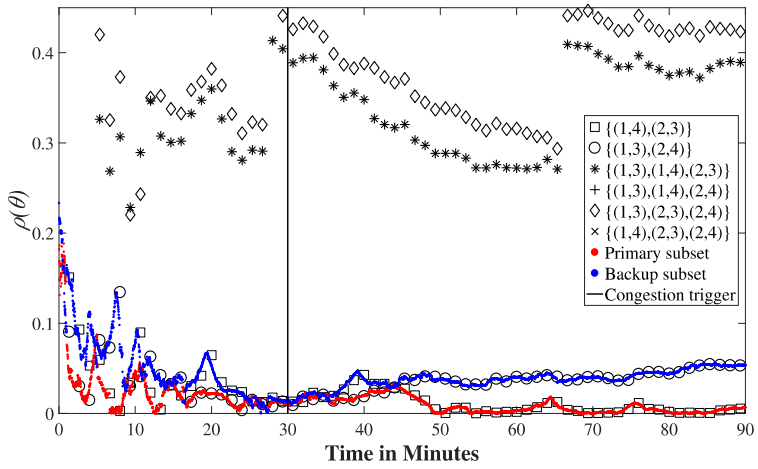


Figure 11.16 $\rho(\theta)$ in the correlation test of the evaluation in the internet environment.

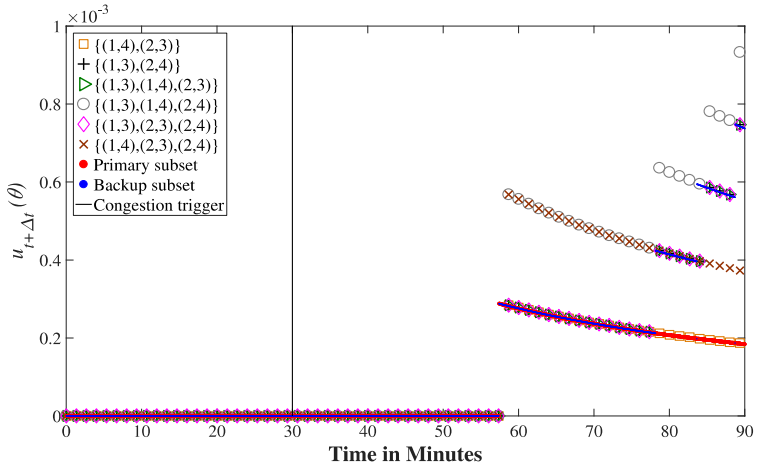


Figure 11.17 $u_{t+\Delta t}(\theta)$ in the correlation test of the evaluation in the internet environment.

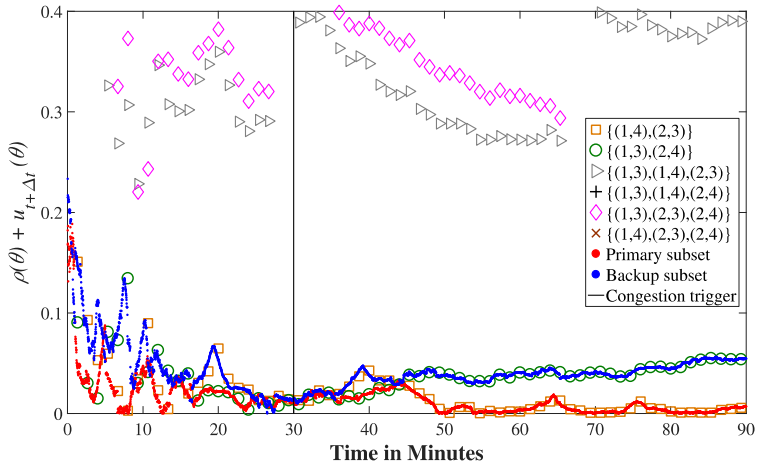


Figure 11.18 The sum $\rho(\theta) + u_{t+\Delta t}(\theta)$ in the correlation test of the evaluation in the internet environment.

model on the paths (*1*,*3*) and (*2*,*4*) was started to change the $\rho(\theta)$ values of the subset {(*1*,*3*),(*2*,*4*)}. Before the congestion model was activated, the subset {(*1*,*3*),(*2*,*4*)} exhibited a lower degree of correlation compared to the subset {(*1*,*4*),(*2*,*3*)}. After the congestion was started, the $\rho(\theta)$ values for {(*1*,*3*),(*2*,*4*)} immediately started to rise. While {(*1*,*3*),(*2*,*4*)} was chosen as θ_{pr} prior to congestion, the periodic reselection adapted its selection to {(*1*,*4*),(*2*,*3*)} within 15 min after starting the congestion model. As the other selection metrics ($u(\theta)$ and $u_{t+\Delta t}(\theta)$) were not impacted in this test, the selection was only based on the correlation degree ($\rho(\theta)$). This can be observed from Figures 11.17 and 11.18 which plots $u_{t+\Delta t}(\theta)$ and the summation $\rho(\theta) + u_{t+\Delta t}(\theta)$. $u(\theta)$ was not plotted as all subsets have 0% unavailability. As shown in Figure 11.17, the subsets selected for θ_{pr} and θ_{ba} have the same MP unavailability probability for about 80 min of the evaluation interval. It is also clear that the summation $\rho(\theta) + u_{t+\Delta t}(\theta)$ for θ_{pr} and θ_{ba} in Figure 11.18 closely follows the trend of $\rho(\theta)$ values in Figure 11.16. This explains why the system choose to select the highly correlated subset ({(*1*,*3*),(*2*,*4*)}) for θ_{ba} after the congestion model was started.

This test showed that the system can detect a shared link and react upon it. Although there was no indication of jointness between the paths within the first 30 minutes, the correlated packet delay patterns caused by the congestion model were successfully detected thereafter and θ_{pr} was reselected accordingly.

3. *Prediction test*

To test the impact of unavailability prediction on MP selection, the *Linux* server was configured using *netem* as follows: The traffic on path (*1*,*3*) was impacted by a random drop (from a normal distribution) of 5% and a drop correlation of 25%. This causes the packet drop to be less random and emulates bursty losses. More specifically, 5% of packets will be dropped, and each successive probability will have a dependency of 25% on the last one such that:

$$Pr_n = 0.25Pr_{n-1} + 0.75Rand\ (\ ) \tag{11.4}$$

where Pr_n is the drop probability for packet n and $Rand()$ is a random number in the range [0,1]. As indicated in [49], such high and correlated drops are associated with temporary connectivity outages or heavy congestions. It is necessary to indicate here that the *netem* documentations clearly state that the correlated drop created using equation (11.4) is rather an approximation than a true statistical correlation. The initial drop on the path (*1*,*3*) is to cause the values of $u_{t+\Delta t}(\theta)$ of the subset {(*1*,*3*),(*2*,*4*)} to be higher at the beginning of the test. As a result, the system will favor the subset {(*1*,*4*),(*2*,*3*)} as it has lower unavailability probability. After 20 min from the start of the test, the drop on the path (*1*,*3*) is removed and a drop of 20% for the traffic on the path (*1*,*4*) is started. This increased the values of $u_{t+\Delta t}(\theta)$ for the subset {(*1*,*4*),(*2*,*3*)}. The target here it to observe system's reaction when a subset with initially low unavailability probability ({(*1*,*4*),(*2*,*3*)}), suddenly experiences degradation. Figure 11.19 shows the

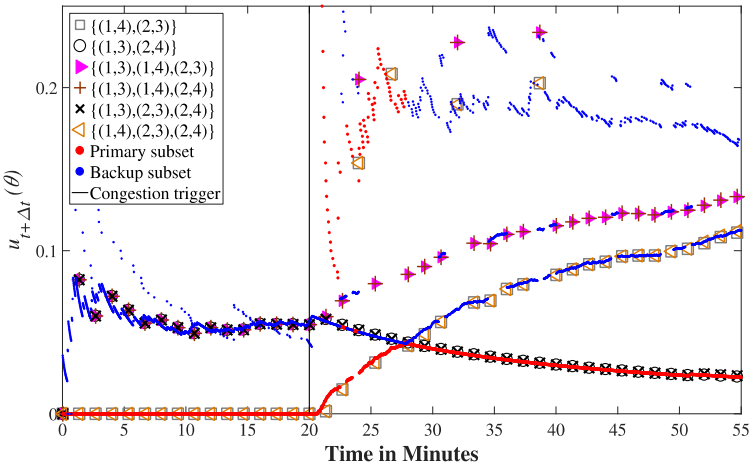


Figure 11.19 $u_{t+\Delta t}(\theta)$ in the prediction test of the evaluation in the internet environment.

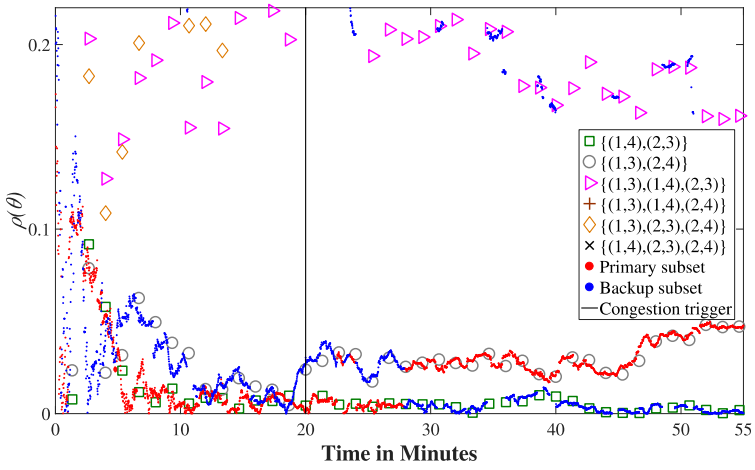


Figure 11.20 $\rho(\theta)$ in the prediction test of the evaluation in the internet environment.

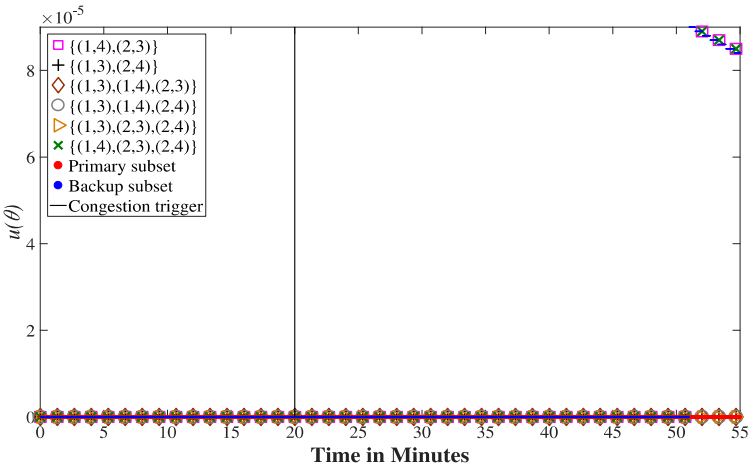


Figure 11.21 $u(\theta)$ in the prediction test of the evaluation in the internet environment.

observed $u_{t+\Delta t}(\theta)$ values by iPRP-RC4CPS and how the MP selection preferred the subset with the lower unavailability probability. Initially the subset $\{(1,3),(2,4)\}$ was not selected for θ_{pr} due to its higher $u_{t+\Delta t}(\theta)$ values and because $\{(1,4),(2,3)\}$ posed a better alternative. After starting the drop on $(1,4)$ indicated by the vertical line in Figure 11.19,

the path experienced severe degradation. This contingency was detected by iPRP-RC4CPS within 10 min and switched its selection of θ_{pr} to the subset $\{(1,3),(2,4)\}$. t_e (Section 11.5.3) was set to 5 to prevent rapid and unnecessary subset switching if there were only short bursty drop on a single path.

Because $u_{t+\Delta t}(\theta)$ always gives the probability of future unavailability depending on the current state, there are always the two probabilities p_{01} or p_{11} that can be sampled after probe transmissions (Section 5.2.2). $u_{t+\Delta t}(\theta)$ returns p_{01} values if the last probe transmissions on the paths of a subset were successful. The corresponding $u_{t+\Delta t}(\theta)$ values are those close to the bottom of the figure. Every time there is a loss of a probe packet, the state of each path changes to 1 in the Gilbert model and $u_{t+\Delta t}(\theta)$ returns the summation that includes one or more p_{11} values. The corresponding $u_{t+\Delta t}(\theta)$ values are the ones close to the top of the figure.

After removing the drop on the path $(1,3)$, a decreasing trend of $u_{t+\Delta t}(\theta)$ values of the subset $\{(1,3),(2,4)\}$ can be observed. This is because the probabilities p_{01} and p_{11} decrease over time if no more packets are lost. By contrast, an increasing trend of $u_{t+\Delta t}(\theta)$ values of the subset $\{(1,4),(2,3)\}$ can be seen after triggering the drop on the path $((1,4))$.

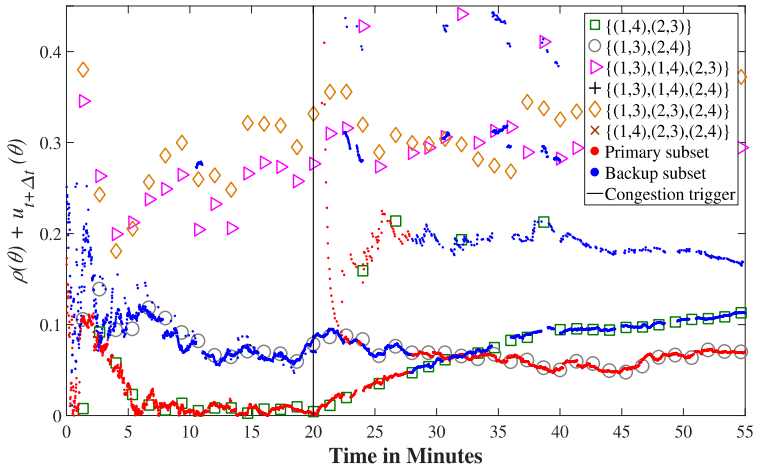


Figure 11.22 The sum $\rho(\theta) + u_{t+\Delta t}(\theta)$ in the prediction test of the evaluation in the internet environment.

$\rho(\theta)$ and $u(\theta)$ for this test are plotted in Figures 11.20 and 11.21 respectively. As it can be seen, $u(\theta)$ for all subset was 0% over almost all the evaluation interval. Moreover, the unavailability constraint u_r was set to 0.01 to not interfere with the selection process and let the system base its subset choice on $\rho(\theta)$ and $u_{t+\Delta t}(\theta)$. On the other hand, $\rho(\theta)$ fluctuates heavily in the first 20 min with lower values than those of $u_{t+\Delta t}(\theta)$. This can be also seen in Figure 11.22, where the subset with the path $(1,3)$ impacted in the first 20 min has higher summation of $\rho(\theta) + u_{t+\Delta t}(\theta)$ and, therefore, was not selected for θ_{pr}. However, this change after the trigger line in the figures. As the subset $\{(1,4),(2,3)\}$ selected for θ_{pr} starts to experience frequent unavailability events, iPRP-RC4CPS switches to $\{(1,3),(2,4)\}$. More specifically, even though that $\{(1,4),(2,3)\}$ has lower

summation of $\rho(\theta) + u_{t+\Delta t}(\theta)$ for a short time after the trigger line in Figure 11.22, it was exchanged with {(*1*,*3*),(*2,4*)}due to its higher unavailability probability. As mentioned previously, iPRP-RC4CPS targets minimizing unavailability, therefore {(*1*,*3*),(*2,4*)} was selected for θ_{pr} shortly after the trigger line.

Figure 11.19 shows how unavailability events on a single path can be identified by the MP selection through $u_{t+\Delta t}(\theta)$. This leads to the conclusion that the use of $u_{t+\Delta t}(\theta)$ is a valid approach to identify temporal unavailability and quantify subsets of paths accordingly.

4. *Unavailability test*

This test aims at illustrating the reliability gains of using iPRP-RC4CPS in terms of the reduced unavailability. The server in this test was configured to cause 2% random drop from a normal distribution for the traffic on each e2e path in the direction from *PC1* to *PC2*. According to [159], the selected drop percentage represent a core network emulation profile with the highest drop severity level. Moreover, the selected drop percentage is higher than the average unavailability percentage of e2e paths captured in previous measurements (assuming that the measured unavailability was mainly caused by packet drop). In addition, two UDP flows from *PC1* to *PC2* were sent from a script representing an application with two different flows. The script sends a pair of packets for the two flows every second each with a size of 64 bytes. The first flow was sent to a monitored port by iPRP-RC4CPS at *PC2* while the second one to an unmonitored port. The first flow was duplicated over the paths of the selected subset for θ_{pr} while the second flow was transmitted over the path (*1*,*3*) in Figure 11.15.

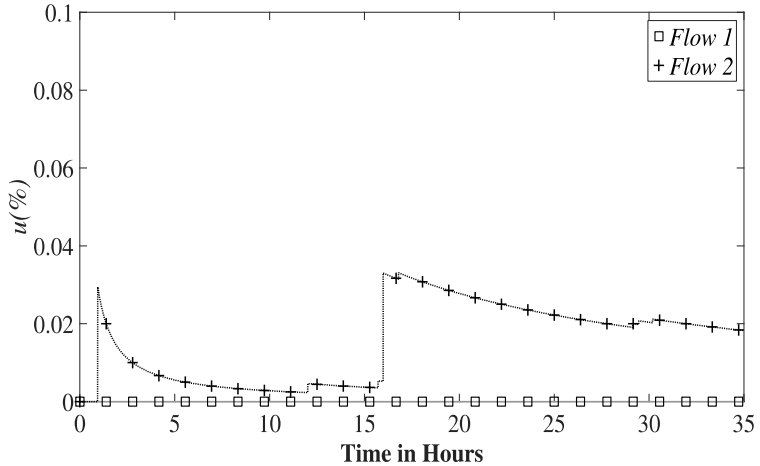


Figure 11.23 Unavailability of the UDP flows, *flow 1* transmitted over iPRP-RC4CPS and *flow 2* over legacy UDP (Internet environment).

In Figure 11.23, the unavailability percentage experienced by the UDP flows calculated using the sequence numbers of received packets over the test interval is depicted. As illustrated, the *flow 2* sent over a single path has a varying unavailability with a maximum value of about 0.04%. The used path for this flow was selected based on

previous monitoring data showing that the selected path has the lowest unavailability compared to the other e2e paths. By contrast, *flow 1* has 0% unavailability during the test interval.

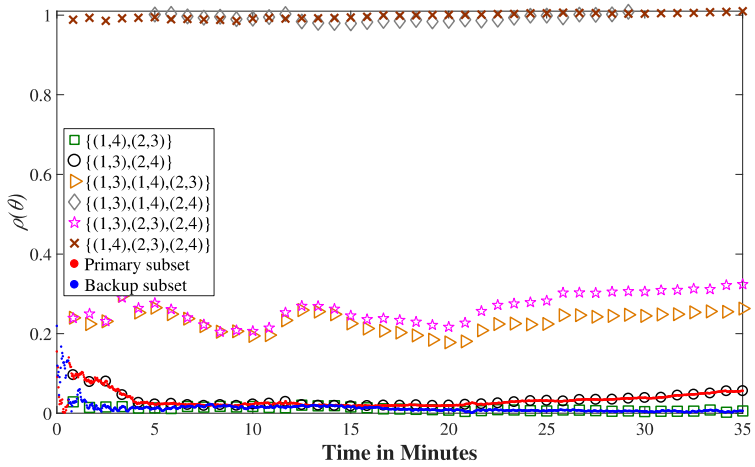


Figure 11.24 $\rho(\theta)$ in the unavailability test of the evaluation in the internet environment.

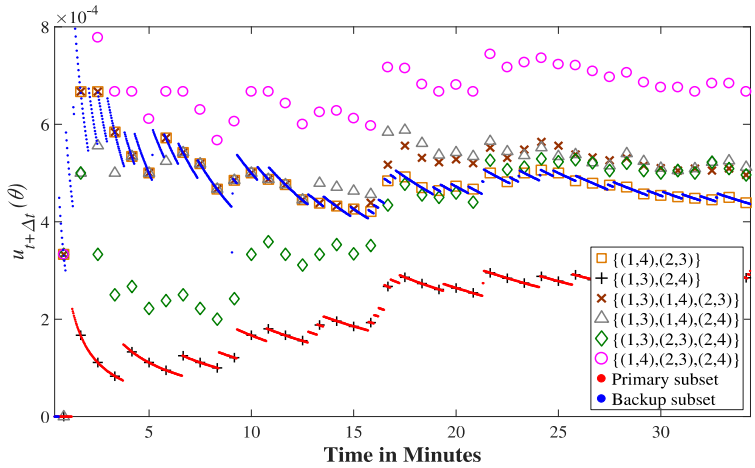


Figure 11.25 $u_{t+\Delta t}(\theta)$ in the unavailability test of the evaluation in the internet environment.

The metrics of the available subsets as measured by iPRP-RC4CPS, Figures 11.24, 11.25, and 11.26 plot $\rho(\theta)$, $u_{t+\Delta t}(\theta)$, and $\rho(\theta) + u_{t+\Delta t}(\theta)$ respectively. $u(\theta)$ was not plotted as it was zero for all subsets during the evaluation interval. As a result, the use of any subset would provide 0% unavailability. In addition, the Figures 11.24, 11.25, and 11.26 show that the selected subsets by iPRP-RC4CPS for θ_{pr} and θ_{ba} have the lowest unavailability probability and the highest diversity (lowest correlation degree as illustrated in Figure 11.24). Even though that the subset $\{(1,4),(2,3)\}$ has lower summation of $\rho(\theta) + u_{t+\Delta t}(\theta)$ during most of the evaluation interval, it was selected as θ_{ba}. This because it has a higher $u_{t+\Delta t}(\theta)$ throughout most of the evaluation interval. As a result, iPRP-RC4CPS and based on the t_e (Section 11.5.3) condition, exchanges θ_{pr} and θ_{ba} immediately after the periodic reselection.

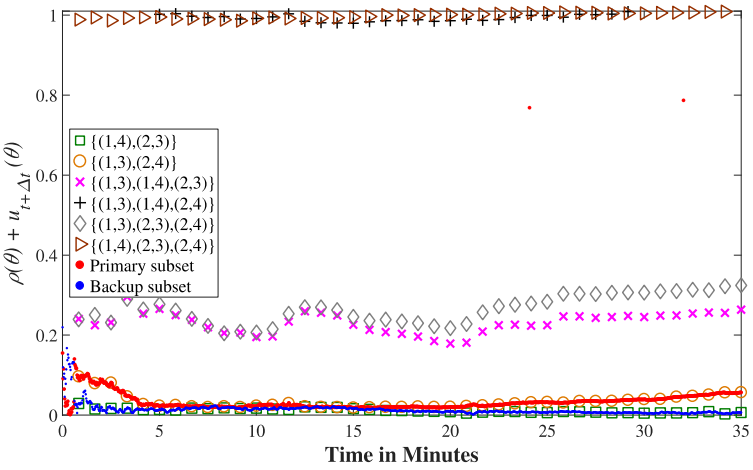


Figure 11.26 The sum $\rho(\theta) + u_{t+\Delta t}(\theta)$ in the unavailability test of the evaluation in the internet environment.

The results in this section, especially the one presented in Figure 11.23, clearly indicate the capability of iPRP-RC4CPS to provide high communication reliability.

12 Conclusion and Future Work

12.1 Conclusion

High communication reliability is one of the main requirements to realize CPSs in critical infrastructures such as smart grids and water distribution systems. As indicated in the literature, low reliability in such infrastructures might result in financial costs and human fatalities. Solutions with this regard have mainly utilized special networks (local control networks) for local area communications or dedicated networks or links (leased lines) for wide area communications. Wide area communications are needed for CPSs that span large geographical areas such as smart grids. A cost effective solutions to realize the communications for such systems is the Internet. However, if the reliability requirements of the different smart grid applications (the considered applications in this dissertation) are compared to the provided reliability by the Internet e2e paths, then the Internet provides inadequate reliability. More specifically, the unavailability requirements of the different smart grid applications can be as low as 0.0001%. By contrast, the measurements indicate that the unavailability of Internet e2e paths is often above 1%. In this dissertation, the problem of improving communication reliability when using the Internet for CPSs was considered.

To address this problem, it was found that redundancy is one of the widely used approaches to improve reliability. The work in the literature indicated also that the core parts of the Internet have a MP nature for reliability and load balancing purposes. In addition, new protocols that support MP communication at the application and transport layers were proposed and standardized. Some of these protocols have sophisticated design that is compatible with middleboxes and can be deployed in the Internet as well as in dedicated IP networks. Such protocols allow multihomed end-systems to use all network interfaces connecting to different access ISPs concurrently. Therefore, it was expected that MP communication using multiple e2e paths attained using different pairs of access ISPs along with data duplication will improve reliability when using the Internet. However, it was necessary to consider the following important issues. First, the redundancy overhead in terms of the number of utilized e2e paths must be minimized. This due to a number of reasons including: (i) reduce costs especially if carriers charge per data volume carried, (ii) improve scalability of the approach for wide deployment in the future, (iii) limit the redundant data by the needed reliability levels to avoid wasting network resources and creating networks congestions, and (iv) decrease overhead at the receiver to handle redundant data packets. Second, the non-transparent and continuously evolving infrastructure of the Internet makes it difficult to estimate the expected improvement in communication reliability when using MP communication. Third, almost all existing MP protocols are throughput oriented and do not give end users control on paths selection. As a result, a number of objectives have been determined and

two main research questions have been defined in Section 2.2. The first research question targeted assessing the potential of MP communication in the case of the Internet. The second research question was regarding the main requirement of minimizing the redundancy overhead and the number of used paths based on the desired reliability by the application.

To answer the first research question, the reliability of different Internet e2e paths as well as their possible subsets between multihomed VNs in different cities in Europe was evaluated. These VNs consisted of two or three end-systems connected to different access ISPs in each city. The reliability was mainly measured in terms of the unavailability of communication service and the diversity of e2e paths. The results of these measurements clearly indicated the support of MP communication for the reliability requirements of all smart grid applications. More specifically, there were different subsets that provided less than 0.0001% of unavailability during the evaluation intervals. In addition, the diversity results indicated the existence of subsets of paths (with 2 or more paths) where all paths traverse completely different networks provided that each path has a different pair of access ISPs. These results further motivated the use of MP communication in the case of the Internet. With regard to the second research question, the approach Reliable MP Communication for CPSs (RC4CPS) was proposed. It is an e2e approach residing at the end-systems which neither require cooperation from networking devices nor the use of additional intermediate devices in the Internet. The approach provides online monitoring and dynamic selection of e2e paths to satisfy the application desired unavailability threshold. For those subsets that fulfill the initial unavailability threshold, the approach uses two additional secondary selection metrics. These are the MP diversity and MP unavailability probability. The diversity measure enables the differentiation between subsets based on the degree of correlation where subsets with low correlation degree between their paths are preferred. In addition, subsets with low MP unavailability probability which reflects the temporal unavailability of the subset's paths are also preferred. The selection based on the secondary metrics is formulated as a minimization problem where any subset with two or more paths is selected when it has the minimum summation. To further boost the reliability gains when using MP communication, it was proposed to have two subsets for data transmission. The first is called the primary subset and the second is called the backup subset. The proposal of the backup subset is to provide a fast alternative for the primary subset if its availability degraded. The online procedures for exchanging the primary and backup subsets as well as for triggering reselections of these subsets (provided in Section 5.4) were proposed to count for short-term and long-term variations in the attributes of the considered subsets.

RC4CPS was implemented two times. The first implementation is done using MATLAB in a multihomed PC and provided an evaluation platform for the approach and its adopted mechanisms without providing data transmission. The second

implementation was done based on the MP transport protocol iPRP. The second implementation is referred to as iPRP-RC4CPS which provides an easy-to-deploy implementation that integrates the RC4CPS features in iPRP and provides data transmission for real world applications. The selection of iPRP for the implementation is done based on a number of requirements drawn from the feature of RC4CPS approach and the technical challenges of today's Internet. The evaluation of the first implementation is carried out using real-world Internet e2e paths while the second implementation was evaluated in both, a controlled lab environment and in the Internet. Evaluations using both implementations showed the effectiveness of adopted mechanism in selecting subsets of paths that adhere to the unavailability threshold desired by the applications as well as selecting subsets that feature the highest diversity and the lowest future unavailability. The obtained results using Internet paths show that there were different subsets that provide unavailability less than 0.0001%. In both implementation, RC4CPS selected subsets with the minimum number of paths and the lowest summations of MP correlation and MP future unavailability for the primary and backup subsets. Moreover, both implementations clearly illustrated the capability of RC4CPS in detecting paths with shared links especially when congestions occur. In addition, the unavailability probability measure, and unlike the average unavailability, provided an instantaneous measure of the actual state of the different paths in a subset. This allowed changing the used subset in a very short time if its paths start to experience frequent unavailability events. With such capability, the unavailability reductions were maximized by using subsets with lower unavailability probability (i.e. low number of unavailability events on the paths).

To conclude, RC4CPS approach proposed in this dissertation can provide the needed availability levels when using the Internet for geographically distributed CPSs such as smart grids. The approach provides the needed mechanisms to perform the online monitoring and dynamic selection of paths to limit the extra redundancy by the specified application's maximum allowed unavailability. Moreover, and through the dynamic online path selection, RC4CPS counts for the varying reliability of the Internet e2e paths. The approach was evaluated using two implementations that confirmed the RC4CPS capability in providing high communication reliability.

12.2 Future Work

In the dissertation, the transport layer implementation, iPRP-RC4CPS, was evaluated in the Internet using local IP addresses and VPN services. This allowed evaluating iPRP-RC4CPS without using public IPs. Another possibility to use iPRP-RC4CPS in the Internet without public IP addresses is through NATs. This in turn requires further tweaking of the iPRP protocol to deal with NATs which perform exchange of local IPs to public IPs. NATs support will allow individuals like researchers to use iPRP-RC4CPS and to adapt it for new use cases. In addition, the use of public IPs by

individuals might not be cost effective or efficient considering the limited number of available IP4 addresses. On the other hand, the use of public IPs by large organizations or utility operators is affordable and even necessary for the different communication purposes in their networks. Hence, the support of NATs by iPRP-RC4CPS would further facilitate the deployment of the protocol in the different environments and for different users.

RC4CPS was designed for unicast mode of communications which is dominant in the case of the Internet. Nevertheless, further support of multicast mode of communications would increase the scalability of the approach. The multicast mode of communication is already supported by the iPRP protocol and, consequently, would not complicate the integration of such additional feature of RC4CPS.

In RC4CPS, additional monitoring data packets are sent in order to assist the different e2e paths. A possible mean to reduce this extra overhead is to utilize the data packets sent over the selected paths (i.e. the paths of θ_{pr} and θ_{ba}). This also can be reduced by utilizing the *iPRP_CAP* messages sent periodically by the iPRP-RC4CPS receiver. However, an important challenge to be addressed with this regard is how to merge the information attained from the monitoring packets and that from data and *iPRP_CAP* packets. More specifically, the different types of packets have, for example, different frequencies. This would impact the granularity of monitoring information on the different paths. Now if the data packets have a higher frequency, this will increase/decrease the unavailability probability of the paths monitored using the data packets. This is mainly due to the larger number of packets considered compared to that of the other paths where no data packets are transmitted (only monitoring packets). In addition, RC4CPS supports online reselection of the paths for data replication, which might result in change of the type of packets transmitted over the paths (monitoring and/or data packets). Regardless of this complexity involved in reducing the monitoring overhead while maintaining the required maximum unavailability, it is an interesting improvement of the RC4CPS approach in the future.

Appendix

Appendix A: Detailed Diversity and Unavailability Results

In this appendix, the detailed diversity and unavailability results briefly introduced in Chapter 6 are provided.

Unavailability Results

The obtained unavailability results to the *Cologne*, *Warsaw*, *Stockholm*, and *Milan* VNs from *Lemgo* VN in *Setup 2* (Section 6.4) are provided in the tables below. For each location of the destination VNs, three tables for the unavailability of 1-, 2-, and 3-path subsets are presented. These results were obtained according to the measurement setup and analysis provided in Chapter 6.

The following 3 tables are for the unavailability measurement between *Lemgo* and *Cologne*.

Table A.1 Unavailability results for each e2e path between *Lemgo* and *Cologne* VNs in S*etup 2* (Section 6.4).

Path	$u(\theta)$ (%)	$u(\theta)$ (s)
(24,3)	0.8751	5481.400
(24,4)	0.8366	5240.290
(24,5)	0.8627	5404.046
(25,3)	0.2373	1486.584
(25,4)	0.2556	1601.054
(25,5)	0.2047	1282.159
(26,3)	1.0852	6797.628
(26,4)	1.0456	6549.509
(26,5)	0.9470	5932.300

Table A.2 Unavailability results for the 2-path subsets between *Lemgo* and *Cologne* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,3),(24,4)}	0.0003	191.928	*{(24,4),(26,3)}*	0.0001	54.025	*{(25,3),(26,4)}*	0.0001	39.392
{(24,3),(24,5)}	0.0005	306.782	*{(24,4),(26,4)}*	0.0001	80.745	*{(25,3),(26,5)}*	0.0000	28.443
{(24,3),(25,3)}	0.0000	10.013	*{(24,4),(26,5)}*	0.0001	45.463	*{(25,4),(25,5)}*	0.0010	614.361
{(24,3),(25,4)}	0.0000	20.009	*{(24,5),(25,3)}*	0.0000	2.598	*{(25,4),(26,3)}*	0.0000	19.939
{(24,3),(25,5)}	0.0000	17.650	*{(24,5),(25,4)}*	0.0000	6.041	*{(25,4),(26,4)}*	0.0000	29.781
{(24,3),(26,3)}	0.0001	72.732	*{(24,5),(25,5)}*	0.0000	3.685	*{(25,4),(26,5)}*	0.0000	17.571
{(24,3),(26,4)}	0.0002	96.945	*{(24,5),(26,3)}*	0.0001	44.046	*{(25,5),(26,3)}*	0.0000	17.304
{(24,3),(26,5)}	0.0000	29.622	*{(24,5),(26,4)}*	0.0002	119.285	*{(25,5),(26,4)}*	0.0000	19.719
{(24,4),(24,5)}	0.0004	247.513	*{(24,5),(26,5)}*	0.0001	69.578	*{(25,5),(26,5)}*	0.0000	24.789
{(24,4),(25,3)}	0.0000	10.079	*{(25,3),(25,4)}*	0.0009	568.942	*{(26,3),(26,4)}*	0.0002	104.350
{(24,4),(25,4)}	0.0000	8.807	*{(25,3),(25,5)}*	0.0009	559.104	*{(26,3),(26,5)}*	0.0001	54.893
{(24,4),(25,5)}	0.0000	14.345	*{(25,3),(26,3)}*	0.0000	21.409	*{(26,4),(26,5)}*	0.0002	120.289

Table A.3 Unavailability results for the 3-path subsets between *Lemgo* and *Cologne* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,3),(24,4),(24,5)}	0.0001	52.877	*{(24,4),(25,4),(26,5)}*	0.0000	0.000
{(24,3),(24,4),(25,3)}	0.0000	0.000	*{(24,4),(25,5),(26,3)}*	0.0000	0.000
{(24,3),(24,4),(25,4)}	0.0000	0.000	*{(24,4),(25,5),(26,4)}*	0.0000	0.000
{(24,3),(24,4),(25,5)}	0.0000	0.000	*{(24,4),(25,5),(26,5)}*	0.0000	0.000
{(24,3),(24,4),(26,3)}	0.0000	0.000	*{(24,4),(26,3),(26,4)}*	0.0000	0.000
{(24,3),(24,4),(26,4)}	0.0000	0.000	*{(24,4),(26,3),(26,5)}*	0.0000	0.000
{(24,3),(24,4),(26,5)}	0.0000	0.000	*{(24,4),(26,4),(26,5)}*	0.0000	0.000
{(24,3),(24,5),(25,3)}	0.0000	0.000	*{(24,5),(25,3),(25,4)}*	0.0000	0.000
{(24,3),(24,5),(25,4)}	0.0000	0.000	*{(24,5),(25,3),(25,5)}*	0.0000	0.000
{(24,3),(24,5),(25,5)}	0.0000	0.000	*{(24,5),(25,3),(26,3)}*	0.0000	0.000
{(24,3),(24,5),(26,3)}	0.0000	0.000	*{(24,5),(25,3),(26,4)}*	0.0000	0.000
{(24,3),(24,5),(26,4)}	0.0000	7.928	*{(24,5),(25,3),(26,5)}*	0.0000	0.000
{(24,3),(24,5),(26,5)}	0.0000	0.000	*{(24,5),(25,4),(25,5)}*	0.0000	0.000
{(24,3),(25,3),(25,4)}	0.0000	10.013	*{(24,5),(25,4),(26,3)}*	0.0000	0.000
{(24,3),(25,3),(25,5)}	0.0000	10.013	*{(24,5),(25,4),(26,4)}*	0.0000	0.000
{(24,3),(25,3),(26,3)}	0.0000	0.000	*{(24,5),(25,4),(26,5)}*	0.0000	0.000
{(24,3),(25,3),(26,4)}	0.0000	0.000	*{(24,5),(25,5),(26,3)}*	0.0000	3.117
{(24,3),(25,3),(26,5)}	0.0000	0.000	*{(24,5),(25,5),(26,4)}*	0.0000	0.000
{(24,3),(25,4),(25,5)}	0.0000	10.013	*{(24,5),(25,5),(26,5)}*	0.0000	0.000
{(24,3),(25,4),(26,3)}	0.0000	0.000	*{(24,5),(26,3),(26,4)}*	0.0000	0.000
{(24,3),(25,4),(26,4)}	0.0000	0.000	*{(24,5),(26,3),(26,5)}*	0.0000	0.000
{(24,3),(25,4),(26,5)}	0.0000	0.000	*{(24,5),(26,4),(26,5)}*	0.0000	0.000
{(24,3),(25,5),(26,3)}	0.0000	0.000	*{(25,3),(25,4),(25,5)}*	0.0009	539.316
{(24,3),(25,5),(26,4)}	0.0000	0.000	*{(25,3),(25,4),(26,3)}*	0.0000	14.214
{(24,3),(25,5),(26,5)}	0.0000	0.000	*{(25,3),(25,4),(26,4)}*	0.0000	22.074
{(24,3),(26,3),(26,4)}	0.0000	0.000	*{(25,3),(25,4),(26,5)}*	0.0000	12.960
{(24,3),(26,3),(26,5)}	0.0000	0.000	*{(25,3),(25,5),(26,3)}*	0.0000	14.187
{(24,3),(26,4),(26,5)}	0.0000	0.000	*{(25,3),(25,5),(26,4)}*	0.0000	12.236
{(24,4),(24,5),(25,3)}	0.0000	0.000	*{(25,3),(25,5),(26,5)}*	0.0000	11.195
{(24,4),(24,5),(25,4)}	0.0000	0.000	*{(25,3),(26,3),(26,4)}*	0.0000	0.000
{(24,4),(24,5),(25,5)}	0.0000	0.000	*{(25,3),(26,3),(26,5)}*	0.0000	4.904
{(24,4),(24,5),(26,3)}	0.0000	0.000	*{(25,3),(26,4),(26,5)}*	0.0000	0.000
{(24,4),(24,5),(26,4)}	0.0000	10.031	*{(25,4),(25,5),(26,3)}*	0.0000	14.187
{(24,4),(24,5),(26,5)}	0.0000	10.032	*{(25,4),(25,5),(26,4)}*	0.0000	12.236
{(24,4),(25,3),(25,4)}	0.0000	0.000	*{(25,4),(25,5),(26,5)}*	0.0000	15.652
{(24,4),(25,3),(25,5)}	0.0000	0.000	*{(25,4),(26,3),(26,4)}*	0.0000	0.000
{(24,4),(25,3),(26,3)}	0.0000	0.000	*{(25,4),(26,3),(26,5)}*	0.0000	4.904
{(24,4),(25,3),(26,4)}	0.0000	0.000	*{(25,4),(26,4),(26,5)}*	0.0000	0.000
{(24,4),(25,3),(26,5)}	0.0000	0.000	*{(25,5),(26,3),(26,4)}*	0.0000	0.000
{(24,4),(25,4),(25,5)}	0.0000	0.000	*{(25,5),(26,3),(26,5)}*	0.0000	4.904
{(24,4),(25,4),(26,3)}	0.0000	0.000	*{(25,5),(26,4),(26,5)}*	0.0000	0.000
{(24,4),(25,4),(26,4)}	0.0000	0.000	*{(26,3),(26,4),(26,5)}*	0.0000	5.002

The tables for *Lemgo* and *Milan* are given below.

Table A.4 Unavailability results for each e2e path between *Lemgo* and *Milan* VNs in S*etup 2* (Section 6.4).

Path	$u(\theta)$ (%)	$u(\theta)$ (s)
(24,16)	0.2906	1820.097
(24,17)	0.7276	4557.596
(25,16)	0.2332	1460.868
(25,17)	0.2364	1480.799
(26,16)	0.4582	2870.323
(26,17)	1.1272	7060.483

TableA.5 Unavailability results for the 2-path subsets between *Lemgo* and *Milan* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,16),(24,17)}	0.0002	103.371	*{(24,17),(25,16)}*	0.0000	0.000	*{(25,16),(26,16)}*	0.0000	0.000
{(24,16),(25,16)}	0.0000	0.000	*{(24,17),(25,17)}*	0.0000	0.000	*{(25,16),(26,17)}*	0.0001	40.734
{(24,16),(25,17)}	0.0000	0.000	*{(24,17),(26,16)}*	0.0000	2.732	*{(25,17),(26,16)}*	0.0000	0.000
{(24,16),(26,16)}	0.0000	8.380	*{(24,17),(26,17)}*	0.0001	63.592	*{(25,17),(26,17)}*	0.0000	22.291
{(24,16),(26,17)}	0.0000	17.727	*{(25,16),(25,17)}*	0.0009	579.033	*{(26,16),(26,17)}*	0.0001	70.104

Table A.6 Unavailability results for the 3-path subsets between *Lemgo* and *Milan* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,16),(24,17),(25,16)}	0.0000	0.000	*{(24,17),(25,16),(25,17)}*	0.0000	0.000
{(24,16),(24,17),(25,17)}	0.0000	0.000	*{(24,17),(25,16),(26,16)}*	0.0000	0.000
{(24,16),(24,17),(26,16)}	0.0000	0.000	*{(24,17),(25,16),(26,17)}*	0.0000	0.000
{(24,16),(24,17),(26,17)}	0.0000	0.000	*{(24,17),(25,17),(26,16)}*	0.0000	0.000
{(24,16),(25,16),(25,17)}	0.0000	0.000	*{(24,17),(25,17),(26,17)}*	0.0000	0.000
{(24,16),(25,16),(26,16)}	0.0000	0.000	*{(24,17),(26,16),(26,17)}*	0.0000	0.000
{(24,16),(25,16),(26,17)}	0.0000	0.000	*{(25,16),(25,17),(26,16)}*	0.0000	0.000
{(24,16),(25,17),(26,16)}	0.0000	0.000	*{(25,16),(25,17),(26,17)}*	0.0000	9.194
{(24,16),(25,17),(26,17)}	0.0000	0.000	*{(25,16),(26,16),(26,17)}*	0.0000	0.000
{(24,16),(26,16),(26,17)}	0.0000	0.000	*{(25,17),(26,16),(26,17)}*	0.0000	0.000

The next tables are for *Lemgo* and *Stockholm*.

Table A.7 Unavailability results for each e2e path between *Lemgo* and *Stockholm* VNs in S*etup 2* (Section 6.4).

Path	$u(\theta)$ (%)	$u(\theta)$ (s)
(24,12)	0.7445	4663.491
(24,13)	0.3859	2416.991
(24,20)	0.7318	4583.835
(25,12)	0.2994	1875.313
(25,13)	0.3020	1891.416
(25,20)	0.2589	1621.778
(26,12)	0.7771	4868.055
(26,13)	0.9246	5791.954
(26,20)	0.4602	2882.798

Table A.8 Unavailability results for the 2-path subsets between *Lemgo* and *Stockholm* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,12),(24,13)}	0.0004	276.118	*{(24,13),(26,12)}*	0.0001	34.369	*{(25,12),(26,13)}*	0.0000	9.945
{(24,12),(24,20)}	0.0007	449.678	*{(24,13),(26,13)}*	0.0000	25.604	*{(25,12),(26,20)}*	0.0000	19.073
{(24,12),(25,12)}	0.0000	29.407	*{(24,13),(26,20)}*	0.0000	8.363	*{(25,13),(25,20)}*	0.0010	625.466
{(24,12),(25,13)}	0.0000	17.150	*{(24,20),(25,12)}*	0.0000	0.000	*{(25,13),(26,12)}*	0.0000	11.460
{(24,12),(25,20)}	0.0000	15.791	*{(24,20),(25,13)}*	0.0000	11.238	*{(25,13),(26,13)}*	0.0001	31.326
{(24,12),(26,12)}	0.0001	39.031	*{(24,20),(25,20)}*	0.0000	9.868	*{(25,13),(26,20)}*	0.0000	5.592
{(24,12),(26,13)}	0.0001	34.864	*{(24,20),(26,12)}*	0.0001	64.592	*{(25,20),(26,12)}*	0.0000	1.498
{(24,12),(26,20)}	0.0000	22.618	*{(24,20),(26,13)}*	0.0001	36.383	*{(25,20),(26,13)}*	0.0000	9.936
{(24,13),(24,20)}	0.0004	267.165	*{(24,20),(26,20)}*	0.0000	24.270	*{(25,20),(26,20)}*	0.0000	9.950
{(24,13),(25,12)}	0.0000	4.045	*{(25,12),(25,13)}*	0.0012	734.752	*{(26,12),(26,13)}*	0.0001	84.658
{(24,13),(25,13)}	0.0000	0.000	*{(25,12),(25,20)}*	0.0012	738.436	*{(26,12),(26,20)}*	0.0000	19.831
{(24,13),(25,20)}	0.0000	9.841	*{(25,12),(26,12)}*	0.0000	16.542	*{(26,13),(26,20)}*	0.0000	19.904

Table A.9 Unavailability results for the 3-path subsets between *Lemgo* and *Stockholm* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,12),(24,13),(24,20)}	0.0002	132.221	*{(24,13),(25,13),(26,20)}*	0.0000	0.000
{(24,12),(24,13),(25,12)}	0.0000	0.000	*{(24,13),(25,20),(26,12)}*	0.0000	0.000
{(24,12),(24,13),(25,13)}	0.0000	0.000	*{(24,13),(25,20),(26,13)}*	0.0000	0.000
{(24,12),(24,13),(25,20)}	0.0000	0.000	*{(24,13),(25,20),(26,20)}*	0.0000	0.000
{(24,12),(24,13),(26,12)}	0.0000	9.176	*{(24,13),(26,12),(26,13)}*	0.0000	1.378
{(24,12),(24,13),(26,13)}	0.0000	10.008	*{(24,13),(26,12),(26,20)}*	0.0000	0.000
{(24,12),(24,13),(26,20)}	0.0000	0.000	*{(24,13),(26,13),(26,20)}*	0.0000	0.000
{(24,12),(24,20),(25,12)}	0.0000	0.000	*{(24,20),(25,12),(25,13)}*	0.0000	0.000
{(24,12),(24,20),(25,13)}	0.0000	0.000	*{(24,20),(25,12),(25,20)}*	0.0000	0.000
{(24,12),(24,20),(25,20)}	0.0000	0.000	*{(24,20),(25,12),(26,12)}*	0.0000	0.000
{(24,12),(24,20),(26,12)}	0.0000	14.057	*{(24,20),(25,12),(26,13)}*	0.0000	0.000
{(24,12),(24,20),(26,13)}	0.0000	0.000	*{(24,20),(25,12),(26,20)}*	0.0000	0.000
{(24,12),(24,20),(26,20)}	0.0000	11.779	*{(24,20),(25,13),(25,20)}*	0.0000	0.000
{(24,12),(25,12),(25,13)}	0.0000	17.150	*{(24,20),(25,13),(26,12)}*	0.0000	0.000
{(24,12),(25,12),(25,20)}	0.0000	11.915	*{(24,20),(25,13),(26,13)}*	0.0000	0.000
{(24,12),(25,12),(26,12)}	0.0000	0.000	*{(24,20),(25,13),(26,20)}*	0.0000	0.000
{(24,12),(25,12),(26,13)}	0.0000	0.000	*{(24,20),(25,20),(26,12)}*	0.0000	0.000
{(24,12),(25,12),(26,20)}	0.0000	0.000	*{(24,20),(25,20),(26,13)}*	0.0000	0.000
{(24,12),(25,13),(25,20)}	0.0000	11.915	*{(24,20),(25,20),(26,20)}*	0.0000	0.000
{(24,12),(25,13),(26,12)}	0.0000	0.000	*{(24,20),(26,12),(26,13)}*	0.0000	0.000
{(24,12),(25,13),(26,13)}	0.0000	0.000	*{(24,20),(26,12),(26,20)}*	0.0000	0.000
{(24,12),(25,13),(26,20)}	0.0000	0.000	*{(24,20),(26,13),(26,20)}*	0.0000	0.000
{(24,12),(25,20),(26,12)}	0.0000	0.000	*{(25,12),(25,13),(25,20)}*	0.0009	593.557
{(24,12),(25,20),(26,13)}	0.0000	0.000	*{(25,12),(25,13),(26,12)}*	0.0000	0.000
{(24,12),(25,20),(26,20)}	0.0000	0.000	*{(25,12),(25,13),(26,13)}*	0.0000	9.926
{(24,12),(26,12),(26,13)}	0.0000	0.000	*{(25,12),(25,13),(26,20)}*	0.0000	5.592
{(24,12),(26,12),(26,20)}	0.0000	0.000	*{(25,12),(25,20),(26,12)}*	0.0000	1.498
{(24,12),(26,13),(26,20)}	0.0000	0.000	*{(25,12),(25,20),(26,13)}*	0.0000	0.000
{(24,13),(24,20),(25,12)}	0.0000	0.000	*{(25,12),(25,20),(26,20)}*	0.0000	5.736
{(24,13),(24,20),(25,13)}	0.0000	0.000	*{(25,12),(26,12),(26,13)}*	0.0000	0.000
{(24,13),(24,20),(25,20)}	0.0000	0.000	*{(25,12),(26,12),(26,20)}*	0.0000	0.000
{(24,13),(24,20),(26,12)}	0.0000	12.880	*{(25,12),(26,13),(26,20)}*	0.0000	0.000
{(24,13),(24,20),(26,13)}	0.0000	0.000	*{(25,13),(25,20),(26,12)}*	0.0000	0.000
{(24,13),(24,20),(26,20)}	0.0000	0.000	*{(25,13),(25,20),(26,13)}*	0.0000	0.000
{(24,13),(25,12),(25,13)}	0.0000	0.000	*{(25,13),(25,20),(26,20)}*	0.0000	5.592
{(24,13),(25,12),(25,20)}	0.0000	3.953	*{(25,13),(26,12),(26,13)}*	0.0000	0.000
{(24,13),(25,12),(26,12)}	0.0000	0.000	*{(25,13),(26,12),(26,20)}*	0.0000	0.000
{(24,13),(25,12),(26,13)}	0.0000	0.000	*{(25,13),(26,13),(26,20)}*	0.0000	0.000
{(24,13),(25,12),(26,20)}	0.0000	0.000	*{(25,20),(26,12),(26,13)}*	0.0000	0.000
{(24,13),(25,13),(25,20)}	0.0000	0.000	*{(25,20),(26,12),(26,20)}*	0.0000	0.000
{(24,13),(25,13),(26,12)}	0.0000	0.000	*{(25,20),(26,13),(26,20)}*	0.0000	0.000
{(24,13),(25,13),(26,13)}	0.0000	0.000	*{(26,12),(26,13),(26,20)}*	0.0000	0.000

Finally, the unavailability results between *Lemgo* and *Warsaw* are as follows.

Table A.10 Unavailability results for each e2e path between *Lemgo* and *Warsaw* VNs in S*etup 2* (Section 6.4).

Path	$u(\theta)$ (%)	$u(\theta)$ (s)
(24,10)	0.7704	4825.857
(24,18)	0.5164	3234.423
(25,10)	0.3394	2125.956
(25,18)	0.3248	2034.295
(26,10)	1.0776	6750.151
(26,18)	0.5561	3483.469

Table A.11 Unavailability results for the 2-path subsets between *Lemgo* and *Warsaw* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,10),(24,18)}	0.0002	132.084	*{(24,18),(25,10)}*	0.0000	22.355	*{(25,10),(26,10)}*	0.0001	39.117
{(24,10),(25,10)}	0.0001	37.709	*{(24,18),(25,18)}*	0.0000	18.856	*{(25,10),(26,18)}*	0.0000	17.577
{(24,10),(25,18)}	0.0000	15.037	*{(24,18),(26,10)}*	0.0000	27.625	*{(25,18),(26,10)}*	0.0000	19.925
{(24,10),(26,10)}	0.0001	39.725	*{(24,18),(26,18)}*	0.0001	66.938	*{(25,18),(26,18)}*	0.0000	12.558
{(24,10),(26,18)}	0.0001	31.867	*{(25,10),(25,18)}*	0.0010	622.233	*{(26,10),(26,18)}*	0.0000	24.475

Table A.12 Unavailability results for the 3-path subsets between *Lemgo* and *Warsaw* VNs in S*etup 2* (Section 6.4).

Subset	$u(\theta)$ (%)	$u(\theta)$ (s)	Subset	$u(\theta)$ (%)	$u(\theta)$ (s)
{(24,10),(24,18),(25,10)}	0.0000	0.000	*{(24,18),(25,10),(25,18)}*	0.0000	9.495
{(24,10),(24,18),(25,18)}	0.0000	0.000	*{(24,18),(25,10),(26,10)}*	0.0000	0.000
{(24,10),(24,18),(26,10)}	0.0000	0.000	*{(24,18),(25,10),(26,18)}*	0.0000	0.000
{(24,10),(24,18),(26,18)}	0.0000	0.000	*{(24,18),(25,18),(26,10)}*	0.0000	0.000
{(24,10),(25,10),(25,18)}	0.0000	1.880	*{(24,18),(25,18),(26,18)}*	0.0000	0.000
{(24,10),(25,10),(26,10)}	0.0000	0.000	*{(24,18),(26,10),(26,18)}*	0.0000	0.000
{(24,10),(25,10),(26,18)}	0.0000	0.000	*{(25,10),(25,18),(26,10)}*	0.0000	0.000
{(24,10),(25,18),(26,10)}	0.0000	2.960	*{(25,10),(25,18),(26,18)}*	0.0000	3.217
{(24,10),(25,18),(26,18)}	0.0000	0.000	*{(25,10),(26,10),(26,18)}*	0.0000	0.000
{(24,10),(26,10),(26,18)}	0.0000	0.000	*{(25,18),(26,10),(26,18)}*	0.0000	0.000

Diversity Results

The obtained Diversity results between the *Frankfurt*, *Cologne*, *Stuttgart*, *Warsaw*, *Stockholm*, *Paris*, and *Milan* VNs in *Setup 2* in Section 6.3 are presented in the following tables. For each location (VN), the diversity results from each node are presented in a separate table. The results in these tables were obtained according to the details and diversity analysis provided in Chapter 6.

The following tables are listed according to the number of the nodes in Figure 6.1.

Table A.13 Diversity of Internet paths from node *1* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(1,2)	Net	6	AS174, AS3320	Cogent, Telekom
(1,3)	Y	9	AS174, AS8422	Cogent, NetCologne
(1,4)	Y	9	AS174, AS1299, AS8365	Cogent, TeliaSonera, MANDA
(1,6)	!Net	4	AS174, *	Cogent, *
(1,7)	Y	6	AS174, AS553	Cogent, BelWue
(1,8)	Net	6	AS174, AS3320	Cogent, Telekom
(1,9)	Y	5	AS174, AS1299	Cogent, TeliaSonera
(1,10)	Y	10	AS174, AS2914	Cogent, NTTCOM
(1,12)	Y	8	AS174, AS1299, AS6939	Cogent, TeliaSonera, Hurricane
(1,13)	Y	10	AS174, AS3356, AS3549	Cogent, Level3
(1,14)	Y	6	AS174, AS6762	Cogent, Seabone
(1,15)	Y	10	AS174, AS1299, AS15412	Cogent, TeliaSonera, GCX
(1,16)	Y	6	AS174, AS6453	Cogent, TATA
(1,17)	Y	7	AS174, AS1239	Cogent, Sprint

Table A.14 Diversity of Internet paths from node *2* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(2,1)	Y	5	AS3320, AS174	Telekom, Cogent
(2,3)	Y	7	AS3320, AS8422	Telekom, NetCologne
(2,4)	Y	6	AS3320, AS8365	Telekom, MANDA
(2,6)	!Net	2	AS3320, *	Telekom, *
(2,7)	Y	9	AS3320, AS33891, AS553	Telekom, Core-Backbone, BelWue
(2,8)	Net	2	AS3320	Telekom
(2,9)	Y	5	AS3320, AS1299	Telekom, TeliaSonera
(2,10)	Y	5	AS3320, AS2914	Telekom, NTTCOM
(2,12)	Y	8	AS3320, AS1299, AS6939	Telekom, TeliaSonera, Hurricane
(2,13)	Y	6	AS3320, AS3549	Telekom, Level3
(2,14)	Y	5/6	AS3320, AS6762	Telekom, Seabone
(2,15)	Y	5	AS3320, AS15412	Telekom, GCX
(2,16)	Y	6	AS3320, AS6453	Telekom, TATA
(2,17)	Y	6	AS3320, AS1239	Telekom, Sprint

Table A.15 Diversity of Internet paths from node *3* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(3,1)	Net	8	AS8422, AS174	NetCologne, Cogent
(3,2)	Net	5	AS8422, AS3320	NetCologne, Telekom
(3,4)	Y	8	AS8422, AS9033, AS8365	NetCologne, ECIX, MANDA
(3,6)	Y	8	AS8422, AS680	NetCologne, DFN
(3,7)	Y	10	AS8422, AS680, AS553	NetCologne, DFN, BelWue
(3,8)	Net	5	AS8422, AS3320	NetCologne, Telekom
(3,9)	Y	7	AS8422, AS1299	NetCologne, TeliaSonera
(3,10)	Y	8	AS8422, AS9057/AS3356, AS2914	NetCologne, Level3, NTTCOM
(3,12)	Y	6	AS8422, AS1200, AS6939	NetCologne, AIE, Hurricane
(3,13)	Y	11	AS8422, AS9057/AS3356, AS3549	NetCologne, Level3
(3,14)	Y	10	AS8422, AS9057/AS3356, AS6762	NetCologne, Level3, Seabone
(3,15)	Y	6	AS8422, AS6695/AS51531, AS15412	NetCologne, DE-CIX, GCX
(3,16)	Y	11	AS8422, AS9057/AS3356, AS6453	NetCologne, Level3, TATA
(3,17)	Y	11	AS8422, AS9057/AS3356, AS1239	NetCologne, Level3, Sprint

Table A.16 Diversity of Internet paths from node *4* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(4,1)	Net	8	AS8365, AS1299, AS174	MANDA, TeliaSonera, Cogent
(4,2)	Net	5	AS8365, AS3320	MANDA, Telekom
(4,3)	Y	8	AS8365, AS8422	MANDA, NetCologne
(4,6)	!Net	8	AS8365, AS35548, *, AS9033	MANDA, smartTERRA, *, ECIX
(4,7)	Y	9	AS8365, AS553	MANDA, BelWue
(4,8)	Net	5	AS8365, AS3320	MANDA, Telekom
(4,9)	Y	6	AS8365, AS1299	MANDA, TeliaSonera
(4,10)	Y	5	AS8365, AS2914	MANDA, NTTCOM
(4,12)	Y	7	AS8365, AS6939	MANDA, Hurricane
(4,13)	Y	11	AS8365, AS2914, AS3356, AS3549	MANDA, NTTCOM, Level3
(4,14)	Y	9	AS8365, AS1299, AS6762	MANDA, TeliaSonera, Seabone
(4,15)	Y	7	AS8365, AS6695/AS51531, AS15412	MANDA, DE-CIX, GCX
(4,16)	Y	7	AS8365, AS1299, AS6453	MANDA, TeliaSonera, TATA
(4,17)	Y	11	AS8365, AS1299, AS1239	MANDA, TeliaSonera, Sprint

Table A.17 Diversity of Internet paths from node *6* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(6,1)	Net	9	AS680, AS174	DFN, Cogent
(6,2)	Net	6	AS680, ASS3320	DFN, Telekom
(6,3)	Y	6	AS680, AS6695/AS51531, AS8422	DFN, DE-CIX, NetCologne
(6,4)	Y	10	AS680, AS9033, AS35548, AS8365	DFN, ECIX, smartTERRA, MANDA
(6,7)	Y	6	AS680, AS553	DFN, BelWue
(6,8)	Net	6	AS680, AS3320	DFN, Telekom
(6,9)	Y	7	AS680, AS9057/AS3356, AS1299	DFN, Level3, TeliaSonera
(6,10)	Y	6	AS680, AS9057/AS3356, AS2914	DFN, Level3, NTTCOM
(6,12)	Y	5	AS680, AS6695/AS51531, AS6939	DFN, DE-CIX, Hurricane
(6,13)	Y	8	AS680, AS9057/AS3356, AS3549	DFN, Level3
(6,14)	Y	7	AS680, AS9057/AS3356, AS6762	DFN, Level3, Seabone
(6,15)	Y	6	AS680, AS6695/AS51531, AS15412	DFN, DE-CIX, GCX
(6,16)	Y	8	AS680, AS9057/AS3356, AS6453	DFN, Level3, TATA
(6,17)	Y	8	AS680, AS9057/AS3356, AS1239	DFN, Level3, Sprint

Table A.18 Diversity of Internet paths from node *7* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(7,1)	Net	5	AS553, AS174	BelWue, Cogent
(7,2)	Net	7	AS553, AS33891, AS3320	BelWue, Core-Backbone, Telekom
(7,3)	Y	8	AS553, AS8422	BelWue, NetCologne
(7,4)	Y	9	AS553, AS8365	BelWue, MANDA
(7,6)	Y	6	AS553, AS680	BelWue, DFN
(7,8)	Net	7	AS553, AS33891, AS3320	BelWue, Core-Backbone, Telekom
(7,9)	Y	6	AS553, AS1299	BelWue, TeliaSonera
(7,10)	Y	7	AS553, AS1299, AS2914	BelWue, TeliaSonera, NTTCOM
(7,12)	Y	7	AS553, AS903, AS6939	BelWue, ECIX, Hurricane
(7,13)	Y	9	AS553, AS9057/AS3356, AS3549	BelWue, Level3,
(7,14)	Y	8	AS553, AS1299, AS6762	BelWue, TeliaSonera, Seabone
(7,15)	Y	10	AS553, AS1299, AS15412	BelWue, TeliaSonera, GCX
(7,16)	Y	8	AS553, AS1299, AS6453	BelWue, TeliaSonera, TATA
(7,17)	Y	10	AS553, AS1299, AS1239	BelWue, TeliaSonera, Sprint

Table A.19 Diversity of Internet paths from node *8* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(8,1)	Y	5	AS3320, AS174	Telekom, Cogent
(8,2)	Net	2	AS3320	Telekom
(8,3)	Y	7	AS3320, AS8422	Telekom, NetCologne
(8,4)	Y	6	AS3320, AS8365	Telekom, MANDA
(8,6)	!Net	2	AS3320, *	Telekom, *
(8,7)	Y	9	AS3320, AS33891, AS553	Telekom, Core-Backbone, BelWue
(8,9)	Y	5	AS3320, AS1299	Telekom, TeliaSonera
(8,10)	Y	5	AS3320, AS2914	Telekom, NTTCOM
(8,12)	Y	8	AS3320, AS1299, AS6939	Telekom, TeliaSonera, Hurricane
(8,13)	Y	6	AS3320, AS3549	Telekom, Level3
(8,14)	Y	6	AS3320, AS6762	Telekom, Seabone
(8,15)	Y	5	AS3320, AS15412	Telekom, GCX
(8,16)	Y	6	AS3320, AS6453	Telekom, TATA
(8,17)	Y	6	AS3320, AS1239	Telekom, Sprint

Table A.20 Diversity of Internet paths from node *9* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(9,1)	Y	5	AS1299, AS174	TeliaSonera, Cogent
(9,2)	Net	3	AS1299, AS3320	TeliaSonera, Telekom
(9,3)	Y	7	AS1299, AS8422	TeliaSonera, NetCologne
(9,4)	Y	6	AS1299, AS8365	TeliaSonera, MANDA
(9,6)	!Net	4	AS1299, AS3356, *	TeliaSonera, Level3, *
(9,7)	Y	5	AS1299, AS553	TeliaSonera, BelWue
(9,8)	Net	3	AS1299, AS3320	TeliaSonera, Telekom,
(9,10)	Y	4	AS1299, AS2914	TeliaSonera, NTTCOM
(9,12)	Y	7	AS1299, AS6939	TeliaSonera, Hurricane
(9,13)	Y	7	AS1299, AS3356, AS3549	TeliaSonera, Level3
(9,14)	Y	5	AS1299, AS6762	TeliaSonera, Seabone
(9,15)	Y	6	AS1299, AS15412	TeliaSonera, GCX
(9,16)	Y	5	AS1299, AS6453	TeliaSonera, TATA
(9,17)	Y	7	AS1299, AS1239	TeliaSonera, Sprint

Table A.21 Diversity of Internet paths from node *10* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(10,1)	Net	3	AS2914, AS174	NTTCOM, Cogent
(10,2)	Net	4	AS2914, AS3320	NTTCOM, Telekom
(10,3)	Y	8	AS2914, AS3356, AS8422	NTTCOM, Level3, NetCologne,
(10,4)	Y	5	AS2914, AS8365,	NTTCOM, MANDA
(10,6)	!Net	4	AS2914, AS3356, *	NTTCOM, Level, *
(10,7)	Y	84	AS2914, AS3356, AS553	NTTCOM, Level3, BelWue
(10,8)	Net	4	AS2914, AS3320	NTTCOM, Telekom
(10,9)	Y	4	AS2914, AS1299	NTTCOM, TeliaSonera
(10,12)	Y	7	AS2914, AS1299, AS6939	NTTCOM, TeliaSonera, Hurricane
(10,13)	Y	8	AS2914, AS3356, AS3549	NTTCOM, Level3
(10,14)	Y	4	AS2914, AS6762	NTTCOM, Seabone
(10,15)	Y	7	AS2914, AS15412	NTTCOM, GCX
(10,16)	Y	5	AS2914, AS6453	NTTCOM, TATA
(10,17)	Net	9	AS2914, AS1239	NTTCOM, Sprint

Table A.22 Diversity of Internet paths from node *12* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(12,1)	Y	10	AS6939, AS1299, AS174	Hurricane, TeliaSonera, Cogent
(12,2)	Net	7	AS6939, AS1299, AS3320	Hurricane, TeliaSonera, Telekom
(12,3)	Y	6	AS6939, AS1200, AS8422	Hurricane, AIX, NetCologne,
(12,4)	Y	7	AS6939, AS1200, AS8365	Hurricane, AIX, MANDA
(12,6)	!Net	2	AS6939, *	Hurricane, *
(12,7)	Y	7	AS6939, AS6695/AS51531, AS553	Hurricane, DE-CIX, BelWue
(12,8)	Net	7	AS6939, AS1299, AS3320	Hurricane, TeliaSonera, Telekom
(12,9)	Y	5	AS6939, AS1299	Hurricane, TeliaSonera
(12,10)	Y	7	AS6939, AS1299, AS2914	Hurricane, TeliaSonera, NTTCOM
(12,13)	Y	6	AS6939, AS3356, AS3549	Hurricane, Level3
(12,14)	Y	5/6	AS6939, AS6762	Hurricane, Seabone
(12,15)	Y	5	AS6939, AS6695/AS51531, AS15412	Hurricane, DE-CIX, GCX
(12,16)	Y	7	AS6939, AS1299, AS6453	Hurricane, TeliaSonera, TATA
(12,17)	Y	9	AS6939, AS1200, AS1239	Hurricane, AIE, Sprint

Table A.23 Diversity of Internet paths from node *13* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(13,1)	Net	9	AS3549, AS3356, AS174	Level3, Cogent
(13,2)	Net	3	AS3549, AS3320	Level3, Telekom
(13,3)	Y	8	AS3549, AS3356, AS8422	Level3, NetCologne
(13,4)	Y	10	AS3549, AS3356, AS2914, AS8365	Level3, NTTCOM, MANDA
(13,6)	!Net	4	AS3549, AS3356, *	Level3, *
(13,7)	Y	8	AS3549, AS3356, AS553	Level3, BelWue
(13,8)	Net	3	AS3549, AS3320	Level3, Telekom
(13,9)	Y	8	AS3549, AS3356, AS1299	Level3, TeliaSonera
(13,10)	Y	7	AS3549, AS3356, AS2914	Level3, NTTCOM
(13,12)	Y	6	AS3549, AS3356, *, AS6939	Level3, *, Hurricane
(13,14)	Y	7	AS3549, AS3356, AS6762	Level3, Seabone
(13,15)	Y	12	AS3549, AS3356, AS1299, AS15412	Level3, TeliaSonera, GCX
(13,16)	Y	8	AS3549, AS3356, AS6453	Level3, TATA
(13,17)	Y	8	AS3549, AS3356, AS1239	Level3, Sprint

Table A.24 Diversity of Internet paths from node *14* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(14,1)	Net	4	AS174	Cogent
(14,2)	Net	5	AS6762, AS3320	Seabone, Telekom
(14,3)	Y	11	AS174, AS8422	Cogent, NetCologne
(14,4)	Y	10	AS6762, AS1299, AS8365	Seabone, TeliaSonera, MANDA
(14,6)	!Net	5	AS6762, AS3356, *	Seabone, Level3, *
(14,7)	Y	9	AS6762, AS3356, AS553	Seabone, Level3, BelWue
(14,8)	Net	5	AS6762, AS3320	Seabone, Telekom
(14,9)	Y	6	AS6762, AS1299	Seabone, TeliaSonera
(14,10)	Y	8	AS6762, AS2914	Seabone, NTTCOM
(14,12)	Y	6	AS6762, AS6939	Seabone, Hurricane
(14,13)	Y	10/11	AS6762, AS3356, AS3549	Seabone, Level3
(14,15)	Y	6	AS6762, AS6695/AS51531, AS15412	Seabone, DE-CIX, GCX
(14,16)	Y	4	AS6762, AS6453	Seabone, TATA
(14,17)	Net	3	AS6762, AS1239	Seabone, Sprint

Table A.25 Diversity of Internet paths from node *15* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(15,1)	Y	7	AS3257, AS174	GTT, Cogent
(15,2)	Net	4	AS15412, AS3320	GCX, Telekom
(15,3)	Y	7	AS15412, AS5459, AS8422	GCX, LINX, NetCologne
(15,4)	!Net	2	AS15412, *	GCX, *
(15,6)	!Net	5	AS3257, AS3356, *	GTT, Level3, *
(15,7)	Y	9	AS15412, AS5459, AS33891, AS553	GCX, LINX, Core-Backbone, BelWue
(15,8)	Net	4	AS15412, AS3320	GCX, Telekom
(15,9)	Y	7	AS15412, AS1299	GCX, TeliaSonera
(15,10)	Y	6	AS2914	NTTCOM
(15,12)	Y	5	AS15412, AS44729, AS6939	GCX, EQUINIX, Hurricane
(15,13)	Y	11/12	AS15412, AS2914, AS3356, AS3549	GCX, NTTCOM, Level3
(15,14)	Y	5	AS15412, AS55459, AS6462	GCX, LINX, Seabone
(15,16)	Y	15/16	AS15412, AS6453	GCX, TATA
(15,17)	Net	7	AS3257, AS1239	GTT, Sprint

Table A.26 Diversity of Internet paths from node *16* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(16,1)	Net	4	AS6453, AS174	TATA, Cogent
(16,2)	Net	4	AS6453, AS3320	TATA, Telekom
(16,3)	Y	8	AS6453, AS3356/AS9057, AS8422	TATA, Level3, NetCologne
(16,4)	Y	8	AS6453, AS2914, AS8365	TATA, NTTCOM, MANDA
(16,6)	!Net	4	AS6453, AS3356, *	TATA, Level3, *
(16,7)	Y	8	AS6453, AS3356, AS553	TATA, Level3, BelWue
(16,8)	Net	4	AS6453, AS3320	TATA, Telekom
(16,9)	Y	5	AS6453, AS1299	TATA, TeliaSonera
(16,10)	Y	5	AS6453, AS2914	TATA, NTTCOM
(16,12)	Y	6	AS6453, AS1299, AS6939	TATA, TeliaSonera, Hurricane
(16,13)	Y	8	AS6453, AS3356, AS3549	TATA, Level3
(16,14)	Y	5	AS6453, AS6762	TATA, Seabone
(16,15)	Y	4	AS6453, AS3257, AS15412	TATA, GTT, GCX
(16,17)	Y	6/8	AS6453, AS1239	TATA, Sprint

Table A.27 Diversity of Internet paths from node *17* in S*etup 2* (Section 6.3).

Path	Destination reached?	# of Hops	ASs	Networks
(17,1)	Net	6	AS1239, AS174	Sprint, Cogent
(17,2)	Net	5	AS1239, AS3320	Sprint, Telekom
(17,3)	Y	10	AS1239, AS3356, AS8422	Sprint, Level3, NetCologne
(17,4)	Y	11	AS1239, AS1299, AS8365	Sprint, TeliaSonera, MANDA
(17,6)	!Net	5	AS1239, AS3356, *	Sprint, Level3, *
(17,7)	Y	10	AS1239, AS3356, AS553	Sprint, Level3, BelWue
(17,8)	Net	5	AS1239, AS3320	Sprint, Telekom
(17,9)	Y	7	AS1239, AS1299	Sprint, TeliaSonera
(17,10)	Y	9	AS1239, AS2914	Sprint, NTTCOM
(17,12)	Y	6	AS1239, AS6695/AS51531, AS6939	Sprint, DE-CIX, Hurricane
(17,13)	Y	9/11	AS1239, AS3356, AS3549	Sprint, Level3
(17,14)	Net	3	AS1239, AS6762	Sprint, Seabone
(17,15)	Y	4	AS1239, AS3257, AS15412	Sprint, GTT, GCX
(17,16)	Y	9	AS1239, AS6453	Sprint, TATA

Appendix B: Detailed Results for the Comparison Test

In this appendix, the detailed results of the Comparison test described in Sections 5.2.1 and 7.4 for all path pairs between *Lemgo* source VN and *Cologne*, *Warsaw*, *Stockholm*, and *Paris* destination VNs are listed in the following tables. These tables were obtained according to the measurements and analyses provided in Chapters 6 and 7. The complete names for the acronyms used in the *Traceroute* tables are given in Table 6.1. As Unitymedia, UPC Austria, and Chello are subsidiaries of Liberty Global Telecommunications Company, their checked IPs belongs to the same ASN (marked with asterisk in the tables). In addition, the *Traceroute* sessions to destination *13* in Table B.7 were unsuccessful and the corresponding cells for ASs and networks where left empty.

In each of the correlation tables, the values of M_x for each pair of paths are provided. The RTT delays used for these calculations are the ones obtained from *Setup 2* during the unavailability evaluation (Section 6.4).

Table B.1 Diversity results between *Lemgo* and *Cologne* VNs (*Setup 2* in Section 6.4) using the *Traceroute* tool.

Path	ASs	Networks
(24,3)	AS3320, AS8422	Telekom, NetCologne
(24,4)	AS3320, AS8365	Telekom, MANDA
(25,3)	AS3209, AS51531, AS8422	Vodafone, DE-CIX, NetCologne
(25,4)	AS3209, AS1200, AS8365	Vodafone, AMSIX, MANDA
(26,3)	AS6830*, AS51531, AS8422	Unitymedia, UPC, DE-CIX, NetCologne
(26,4)	AS6830*, AS51531, AS8365	Unitymedia, UPC, DE-CIX, MANDA

Table B.2 M_x values for the 2-path subsets between *Lemgo* and *Cologne* VNs (*Setup 2* in Section 6.4).

Subset	M_x	Subset	M_x	Subset	M_x
{(24,3),(24,4)}	0.3118	*{(24,4),(25,3)}*	-0.0017	*{(25,3),(26,3)}*	0.0697
{(24,3),(25,3)}	0.0640	*{(24,4),(25,4)}*	0.1045	*{(25,3),(26,4)}*	0.0003
{(24,3),(25,4)}	-0.0040	*{(24,4),(26,3)}*	0.0004	*{(25,4),(26,3)}*	-0.0005
{(24,3),(26,3)}	0.0691	*{(24,4),(26,4)}*	0.1129	*{(25,4),(26,4)}*	0.1053
{(24,3),(26,4)}	0.0016	*{(25,3),(25,4)}*	0.6669	*{(26,3),(26,4)}*	0.0248

Table B.3 Diversity results between *Lemgo* and *Paris* VNs (*Setup 2* in Section 6.4) using the *Traceroute* tool.

Path	ASs	Networks
(24,14)	AS3320, AS6762	Telekom, Sparkle
(24,22)	AS3320, AS1299, AS29075, AS13273	Telekom, Telianet, IELO, TeaserNet
(25,14)	AS3209, AS1273, AS6762	Vodafone, CW, Sparkle
(25,22)	AS3209, AS1299, AS29075, AS13273	Vodafone, Telianet, IELO, TeaserNet
(26,14)	AS6830*, AS6762	Unitymedia, UPC, Chello, Sparkle
(26,22)	AS6830*, AS6453, AS29075, AS13273	Unitymedia, UPC, TATA, Neuronnexion, TeaserNet

Table B.4 M_x values for the 2-path subsets between *Lemgo* and *Paris* VNs (*Setup 2* in Section 6.4).

Subset	M_x	Subset	M_x	Subset	M_x
{(24,14),(24,22)}	0.3238	*{(24,22),(25,14)}*	-0.0001	*{(25,14),(26,14)}*	0.1003
{(24,14),(25,14)}	0.1011	*{(24,22),(25,22)}*	-0.0018	*{(25,14),(26,22)}*	-0.0018
{(24,14),(25,22)}	-0.0040	*{(24,22),(26,14)}*	0.0014	*{(25,22),(26,14)}*	-0.0018
{(24,14),(26,14)}	0.1060	*{(24,22),(26,22)}*	0.0086	*{(25,22),(26,22)}*	0.0009
{(24,14),(26,22)}	0.0032	*{(25,14),(25,22)}*	0.6827	*{(26,14),(26,22)}*	0.0215

Table B.5 Diversity results between *Lemgo* and *Warsaw* VNs (*Setup 2* in Section 6.4) using the *Traceroute* tool.

Path	ASs	Networks
(24,10)	AS3320, AS2914	Telekom, NTTCOM
(24,18)	AS3320, AS3356	Telekom, dhosting
(25,10)	AS3209, AS1273, AS2914	Vodafone, CW, NTTCOM
(25,18)	AS3209, AS51531, AS24724	Vodafone, DE-CIX, ATMAN,
(26,10)	AS6830*, AS9033, AS2914	Unitymedia, UPC, ECIX, NTTCOM
(26,18)	AS6830*, AS3356	Unitymedia, UPC, Level 3

Table B.6 M_x values for the 2-path subsets between *Lemgo* and *Warsaw* VNs (*Setup 2* in Section 6.4).

Subset	M_x	Subset	M_x	Subset	M_x
{(24,10),(24,18)}	0.3468	*{(24,18),(25,10)}*	-0.0017	*{(25,10),(26,10)}*	0.0045
{(24,10),(25,10)}	-0.0044	*{(24,18),(25,18)}*	0.2739	*{(25,10),(26,18)}*	0.0021
{(24,10),(25,18)}	-0.0086	*{(24,18),(26,10)}*	-0.0031	*{(25,18),(26,10)}*	0.0052
{(24,10),(26,10)}	0.0038	*{(24,18),(26,18)}*	0.3047	*{(25,18),(26,18)}*	0.2897
{(24,10),(26,18)}	-0.0058	*{(25,10),(25,18)}*	0.6714	*{(26,10),(26,18)}*	0.0121

Table B.7 Diversity results between *Lemgo* and *Stockholm* VNs (*Setup 2* in Section 6.4) using the *Traceroute* tool.

Path	ASs	Networks
(24,12)	AS3320, AS1299, AS6939	Telekom, Telianet, Hurricane
(24,13)		
(24,20)	AS3320, AS1299, AS2603,	Telekom, Telianet, NORDUnet
(25,12)	AS3209, AS1200, AS6939	Vodafone, AMSIX, Hurricane
(25,13)		
(25,20)	AS3320, AS1200, AS2603	Vodafone, AMSIX, NORDUnet
(26,12)	AS68301, AS51531, AS6939	Unitymedia, UPC, DE-CIX, Hurricane
(26,13)		
(26,20)	AS68301, AS5459, AS2603	Unitymedia, UPC, LINX, NORDUnet

Table B.8 M_x values for the 2-path subsets between *Lemgo* and *Stockholm* VNs (*Setup 2* in Section 6.4).

Subset	M_x	**Subset**	M_x	**Subset**	M_x
{(24,12),(24,13)}	0.5179	*{(24,13),(26,12)}*	0.0018	*{(25,12),(26,13)}*	-0.0030
{(24,12),(24,20)}	0.4962	*{(24,13),(26,13)}*	0.0140	*{(25,12),(26,20)}*	0.0026
{(24,12),(25,12)}	-0.0035	*{(24,13),(26,20)}*	-0.0019	*{(25,13),(25,20)}*	0.9313
{(24,12),(25,13)}	-0.0003	*{(24,20),(25,12)}*	-0.0030	*{(25,13),(26,12)}*	-0.0028
{(24,12),(25,20)}	-0.0046	*{(24,20),(25,13)}*	-0.0018	*{(25,13),(26,13)}*	0.0077
{(24,12),(26,12)}	0.0097	*{(24,20),(25,20)}*	-0.0029	*{(25,13),(26,20)}*	0.0041
{(24,12),(26,13)}	0.0055	*{(24,20),(26,12)}*	0.0068	*{(25,20),(26,12)}*	-0.0028
{(24,12),(26,20)}	-0.0021	*{(24,20),(26,13)}*	0.0026	*{(25,20),(26,13)}*	-0.0023
{(24,13),(24,20)}	0.5197	*{(24,20),(26,20)}*	-0.0027	*{(25,20),(26,20)}*	0.0047
{(24,13),(25,12)}	-0.0103	*{(25,12),(25,13)}*	0.9542	*{(26,12),(26,13)}*	0.0584
{(24,13),(25,13)}	0.0016	*{(25,12),(25,20)}*	0.9392	*{(26,12),(26,20)}*	0.0425
{(24,13),(25,20)}	-0.0108	*{(25,12),(26,12)}*	0.0005	*{(26,13),(26,20)}*	0.0365

Appendix C: Algorithms

In this appendix, the algorithms of the different functions of iPRP as well as those of iPRP-RC4CPS are presented. Even though that the iPRP algorithms can be found in [160] or drawn from the implementation in [147], they are presented in this appendix for completeness of view and to provide a self-contained material.

Algorithm C-1 Ping-Receive Routine (Receiver)

```
1:  create and configure sockets;
2:  while true do
3:      poll ping-receive socket on iPRP ping port;
4:      if packet is received then
5:          send packet back to sender address;
6:      end if
7:  end while
```

Algorithm C-2 Ping-Send Routine for path P_{ij} (Sender)

```
1:  create and configure sockets;
2:  signal PCR that initialization is finished;
3:  while true do
4:      wait for trigger from PCR;
5:      send probe packet (Probe) on ping-send socket (PSS);
6:      while Probe was not returned or did not timeout do
7:          poll PSS for return of Probe;
8:          discard delayed Probes using Probe.SN for identification;
9:      end while
10:     if Probe timed out then
11:         P_ij.Offline = true;                    // Declare path as unavailable
12:         update P_ij.p_11 with equation (7.3);
13:     else if Probe did not time out then
14:         if P_ij.Offline then
15:             P_ij.Offline = false;               // Declare path as available
16:         end if
17:         update P_ij.p_01 with equation (7.2);
18:         update last round-trip time delay P_ij.RTT;
19:     end if
20:     signal PCR that its ready;
21: end while
```

Algorithm C-3 Ping-Control Routine (Sender)

$k \in [0, n]$, with n being the total number of subsets.

```
1:  wait until all PSRs and SR finished initialization;
2:  while true do
3:      signal send trigger to all PSRs and SR;
4:      wait until all PSRs and SR are ready;
5:      if all paths of the subset Sk are unavailable then
6:          Sk.Offline = true;                          // Declare subset as unavailable
7:      Else
8:          Sk.Offline = false;                         // Declare subset as available
9:      end if
10:     for all k subsets do
11:         update average unav. (Sk.Unav) with equation (5.4) using Pij.Offline;
12:         update unav. prediction (Sk.Pred) with equation (5.9) using Pij.p01/11
13:         if all paths in all subsets are available then
14:             update correlation (Sk.Cor) with equations (5.5) and (5.6) using Pij.RTT;
15:         else
16:             keep old values for Sk.Cor;
17:         end if
18:     end for
19:     Probe.SN++;
20:     sleep T_PING;
21: end while
```

Algorithm C-4 Soft-state maintenance (Receiver)

```
1:  while true do
2:      remove inactive devices from the list of active senders;
3:      (last-seen timer expired);
4:      for every packet received on one of the monitored ports or on iPRP data port do
5:          check if the source is in the list of active senders;
6:          if yes then
7:              update associated last-seen timer;
8:          else
9:              put sender in the list of active senders;
10:         end if
11:     end for
12: end while
```

Algorithm C-5 iPRP capability advertisement (Receiver)

```
1:  while true do
2:      send iPRP CAP messages to all devices in the list of active senders;
3:      sleep T_CAP;
4:  end while
```

Algorithm C-6 iPRP session maintenance (Sender)

```
1:  while true do
2:      remove aged entries from the peer-base;
3:      for every received iPRP_CAP message do
4:          if there is no iPRP session established with the destination then
5:              establish iPRP session by creating new entry in the peer-base;
6:              send iPRP_ACK message;
7:          else
8:              update the keep-alive timer;
9:              update peer-base;
10:         end if
11:     end for
12: end while
```

Algorithm C-7 Packet replication (Sender)

```
1:  for every outgoing UDP packet do
2:      load the peer-base;
3:      if there exists an iPRP session that corresponds to the destination socket then
4:          replicate the payload;
5:          append iPRP headers;
6:          send packet copies;
7:      else
8:          forward the packet unchanged;
9:      end if
10: end for
```

Algorithm C-8 Duplicate discard (Receiver)

```
1:  for every packet received on iPRP data port do
2:      get packet sequence number (Packet.SN);
3:      find receiver link (RecvLink) for current packet source;
4:      if no corresponding RecvLink is found then
5:          create RecvLink with iPRP header information;
6:          forward packet to application;          // Sender seen first time, so packet is fresh
7:      else
8:          if isFreshPacket (Packet, RecvLink) then
9:              remove iPRP header;
10:             reconstruct original packet;
11:             forward packet to application;
12:         else
13:             silently discard the packet;
14:         end if
15:     end if
16: end for
```

Algorithm C-9 Fresh packet function (Receiver)

Function to determine whether a packet is fresh, a duplicate, late or too late.

```
1:  function isFreshPacket (Packet, RecvLink)
2:  if Packet.SN equal RecvLink.HighSN then
3:      return false;                                   // Duplicate packet
4:  else if Packet.SN greater RecvLink.HighSN then
5:      put SNs [RecvLink.HighSN + 1, Packet.SN − 1]
6:      in RecvLink.ListSN;                             // Track late packets if present
7:      remove the smallest SNs until RecvLink.ListSN has
8:      MaxLost entries;                                // Clear list of too late packets
9:      RecvLink.HighSN ← Packet.SN;
10:     return true;
11: else if RecvLink.HighSN − Packet.SN greater MaxLost then
12:     return false;                                   // Too late packet
13: else if Packet.SN is in RecvLink.ListSN then
14:     remove Packet.SN from RecvLink.ListSN;
15:     return true;                                    // Late packet
16: Else
17:     return false;                                   // Duplicate late packet
18: end if
```

Algorithm C-10 Selection Routine (Sender)

```
1:  signal PCR that initialization is finished;
2:  while true do
3:      wait for trigger from PCR;
4:      Reselection.Counter++;
5:      if there is a primary subset (Spr) selected then
6:          if there is no backup subset (Sba) selected then
7:              if Spr.Offline then
8:                  subsetselection(Spr, Sba);
9:                  send new Spr to ICD;
10:                 reset Reselection.Counter;
11:             else if Spr.Unav greater MAX_UNAV then
12:                 subsetselection(Spr, Sba);
13:                 send new Spr to ICD;
14:                 reset Reselection.Counter;
15:             end if
16:         else                                    // If there is a backup subset selected
17:             if Spr.Offline and Sba.Offline then
18:                 subsetselection(Spr, Sba);
19:                 send new Spr to ICD;
20:                 reset Reselection.Counter;
21:             else if Spr.Offline and !Sba.Offline then
22:                 if Sba.Unav greater MAX_UNAV then
23:                     subsetselection(Spr, Sba);
24:                     send new Spr to ICD;
25:                     reset Reselection.Counter;
26:                 else
27:                     exchange Spr and Sba;
28:                     send new Spr to ICD;
29:                     reduce Reselection.Counter;     // To search for new Sba sooner
30:                 end if
31:             else if Spr.Unav greater MAX_UNAV then
32:                 if Sba.Offline or Sba.Unav greater MAX_UNAV then
33:                     subsetselection(Spr, Sba);
34:                     send new Spr to ICD;
35:                     reset Reselection.Counter;
36:                 else
37:                     exchange Spr and Sba;
38:                     send new Spr to ICD;
39:                     reduce Reselection.Counter;
40:                 end if
41:             else if Spr.Pred greater Sba.Pred then
42:                 if !Sba.Offline and Sba.Unav less MAX_UNAV then
43:                     if Pred.Counter expired then  // To prevent overreaction during short bursts
```

Algorithm C-10 Selection Routine (Sender)

```
                        exchange S_pr and S_ba;
```

continued on the next page

```
                        reduce Reselection.Counter;
                        send new S_pr to ICD;
                        reset Pred.Counter;
                    else
                        Pred.Counter++;
                    end if
                else
                    reset Pred.Counter;
                end if
            end if
        else
            Reselection.Counter = RESEL_INTV;  // Prepone reselection if no S_pr was selected
        end if
        if Reselection.Counter greater or equal RESEL_INTV then
            subsetselection(S_pr, S_ba);
            if S_pr changed then
                send new S_pr to ICD;
            end if
            reset Reselection.Counter;
        end if
        signal PCR that its ready;
    end while
```

Algorithm C-11 Subset selection function (Sender)

Function to select the primary and backup subset.
$k \in [0, n]$, with n being the total number of subsets with 2- and 3-paths that fulfill the last two conditions in equation (5.14).

```
1:  function subsetselection(Spr, Sba)
2:  for every subset Sk do
3:      if Sk has at least 2 active paths and Sk.Unav less or equal MAX_UNAV then
4:          if Sk.Cor + Sk.Pred is lowest so far then
5:              Spr = Sk;
6:          end if
7:      end if
8:  end for
9:  if no Spr was found then
10: for every subset Sk do
11:     if Sk has at least 1 active path and Sk.Unav less or equal MAX_UNAV then
12:         if Sk.Cor + Sk.Pred is lowest so far then
13:             Spr = Sk;
14:         end if
15:     end if
16: end for
17: if a Spr was found then
18:     for every subset Sk do
19:         if !Sk.Offline and Sk not equal Spr and Sk.Unav less or equal MAX_UNAV then
20:             if Sk has at least 2 active paths and has at least 2 paths that are not in Spr then
21:                 if Sk.Cor + Sk.Pred is lowest so far then
22:                     Sba = Sk;
23:                 end if
24:             end if
25:         end if
26:     end for
27:     if no Sba was found then
28:         for every subset Sk do
29:             if !Sk.Offline and Sk not equal Spr and Sk.Unav less or equal MAX_UNAV then
30:                 if Sk has at least 1 active path and has at least 2 paths that are not in Spr then
31:                     if Sk.Cor + Sk.Pred is lowest so far then
32:                         Sba = Sk;
33:                     end if
34:                 end if
35:             end if
36:         end for
37:     end if
38: end if
39:  if Spr.Pred greater Sba.Pred then      // For the case if cor. + pred. is lower on Spr..
40:     exchange Spr and Sba;                //..but pred. alone is lower on Sba
41: end if
42: return Spr and Sba;
```

Appendix D: iPRP-RC4CPS Files and Functions

In this appendix, the source files of iPRP-RC4CPS as well as the different functions included in them are described in a tabular form. The first column in the following table lists the daemons association of the main source files (listed in the 2nd column) which contain different functions and routines (provided in the 3rd column). The source files are followed by the utility files which provide the functions required by the different routines and do not belong to a certain daemon. Therefore, these are associated with the iPRP-RC4CPS library as they are available for all daemons.

The header files used in the implementation and their contents are not provided here.

Table D.1 Documentation of all code files of iPRP-RC4CPS and their function.

Daemon association	File name	Function name	Function description
ICD	**icd.c**	*main()*	Provides the environment needed for the ICD routines. It loads the available interfaces and their INDs from a file before establishing the control sockets needed to receive control messages sent by other hosts. It also sets up the required pipes to send messages between the different daemons.
		control_routine()	The Control Routine monitors the iPRP control port such that any received *iPRP_CAP* messages are forwarded to the *send_routine()*.
		receiver_ports_routine()	It reads the file that contains the monitored ports or updates the monitored ports and deletes the old ones in case that the ports in the file were changed. It also sets up an iptables rule for the monitored ports with NFQUEUE. It starts the IRD, IMD and IPD when a valid working port is monitored. It also forwards the ID of the NFQUEUE queue to the IMD in order to access the queue and handle the packets intercepted using the queue.
		receiver_sendcap_routine()	The Send-CAP Routine first creates a socket to transmit messages. Then, it sends *iPRP_CAP* messages periodically every T_{CAP} seconds targeting all senders currently available in the active senders list.
		sender_routine()	This routine receives *iPRP_CAP* messages forwarded by the Control Routine and processes them as expected. It uses these messages to create or update the corresponding peer-bases and creates, for new receivers, the ISDs required to replicate the outgoing traffic. It also obtains the new path configurations provided by the IPD, starts the peer-base updates correspondingly, and signals the changes to the ISD.
	receiver.c	*get_active_senders()*	Gets the active senders by calling the *activesenders_load()* function.
		send_cap()	Constitutes and sends *iPRP_CAP* messages.
		get_monitored_ports()	Retrieves the ports to be monitored from a file.
		get_interfaces()	Reads the IP addresses of the available interfaces and their associated INDs from a file.
	sender.c	*sender_init()*	Initiates the current sender links list.
		peerbase_query()	Perform inquiries regarding matching sender links for a given Source port, destination port, and destination address in the peer-bases.
		return_sender_links()	Given a destination address, this function provides all peer-base sender links.

Daemon association	File name	Function name	Function description
		peerbase_insert()	Creates or updates a peer-base. This is done by adding/updating the sender link and also storing the peer-base in a file. The updating of peer-bases is triggered by a path reconfiguration where an old peer-base is retrieved and updated using the new path configuration.
		peerbase_update()	The peer-base alive timer is updated using this function.
		get_selectedpaths()	Performs the initial path configuration by reading the selected paths from a file.
		get_iface_from_ind()	Given an IND, the function finds the associated interface for it in the host structure.
		get_queue_number()	Provides new queue number that is not already used for an NFQUEUE rule.
IMD	**imd.c**	*main()*	Prepares the environment for the routines in the IMD daemon.
		monitor_routine()	Checks for incoming packets to the monitored application port and use the *handle_packet()* function to handle them. The interception of the application data packets is done using the NFQUEUE which sends them to this routine.
		cleanup_routine()	Removes old entries in the active senders list and writes the changes down to the file. The routine uses the $T_{IMD\_CLEANUP}$ timer where senders that do not send data packets before the timer expiration are removed and *iPRP_CAP* messages to it are stopped. Once new messages arrive, an entry for the sender is added again and the transmission of CAP messages is continued.
		handle_packet()	This callback function is part of the NFQUEUE and is called for every packet queued in the *monitor_routine()*. With each received packet, the function updates the corresponding sender expiration timer in the active senders list. It creates an active sender entry if it is the first packet it sees from this sender.
	util.c	*activesenders_find_entry()*	Looks for entries in the active senders cache using a given source address, source port, and destination port.
		activesenders_create_entry()	Adds entries to the active senders using a given source address/port and the destination address/port.
IRD	**ird.c**	*main()*	Make the environment ready for the IRD routines by setting up and configuring iptables rules using NFQUEUE to manipulate the handling of UDP packets.
		receive_routine()	Intercepts packets going to an iPRP monitored port using NFQUEUE targets created in the iptables. It uses the *handle_packet()* function to manipulate how those packets are handled.
		cleanup_routine()	Deletes receiver link if no data packets are received within $T_{IRD\_CLEANUP}$
		handle_packet()	For each queued packet, this callback function, which part of the NFQUEUE, is called. It reconstructs each queued packet to its original form by extracting the payload and uses it in a new packet that has the original UDP/IP header (with the source and destination information obtained from the iPRP header). The packet is forwarded after removing the iPRP header to the IMD.
	util.c	*receiver_link_get()*	It finds the receiver link given the SNSID of the iPRP header.
		receiver_link_create()	Given an iPRP header, the function creates the receiver link structure.
		is_fresh_packet()	Is the duplicate-discard algorithm and decides whether the received packet will be forwarded to the application (if it is a fresh packet) or be dropped.
		ip_checksum()	Given an IP header, the function computes the checksum for it.
		udp_checksum()	Given a UDP header and the corresponding IP pseudo-header, the function calculates the UDP checksum.

Daemon association	File name	Function name	Function description
ISD	**isd.c**	*main()*	Sets up the environment required for the ISD routines and configures the iptables rules using NFQUEUE to control the UDP packet handling.
		send_routine()	The routine utilizes the NFQUEUE targets created in the iptables in order to intercept packets targeting the monitored port by iPRP-capable receiver. This allows manipulating the handling of the queued packets using the *handle_packet()* function.
		config_routine()	Configures the send sockets according to the peer-base that was loaded initially. It waits for path reconfiguration signaled by the ICD to retrieve the peer-base again and to reconfigure the send sockets correspondingly.
		handle_packet()	This callback function is part of the NFQUEUE and is called for each queued packet. The function extracts the payload of each packet and uses it in multiple new empty packets with modified UDP/IP header with source and destination information based on the current path configuration (based on the selected e2e paths). After that it adds the header of iPRP to the payload. Then it sends the packets (copies of original packet) over the selected paths.
IPD	**ipd.c**	*main()*	Prepares the environment for the IPD routines including the ping-receive, -send, -control routines. It also sets up all the needed path and subset structures and provides all possible subsets with 2 and 3 paths using the current path configuration.
		pingreceive_routine()	The routine creates a thread for each individual active network interface. Each thread monitors the corresponding interface for probe messages and sends them back to the sender interface.
		pingsend_routine()	The routine creates an individual thread for each e2e path to the destination. It transmitts probe messages and waits to receive the corresponding acknowledgments. In addition, the RTT delays as well as timeouts are logged. The *pingcontrol_routine()* synchronize all threads to send at the same time and also controls the amount of sent pings. The routine calculates and stores also the unavailability intervals for the monitored paths. With each iteration, the average unavailability as well as the unavailability probability are also calculated. The ping threads are launched only when the local machine becomes a sender (after the ISD is started and received the path configuration (sender link) from ICD via a pipe in the *main()* function)
		pingcontrol_routine()	It clocks the *pingsend_routine()* threads as well as the *selection_routine()* thread. It also calculates $u(\theta)$, $u_{t+\Delta t}(\theta)$, and $\rho(\theta)$ of all path subsets.
		selection_routine()	Selects the θ_{pr} and θ_{ba} periodically. It also instantly exchanges θ_{pr} and θ_{ba} as described in Section 5.4 if θ_{pr} reliability is degraded. The newly selected θ_{pr} is sent to the ICD using *subset_reconfiguration()* function.
		subset_reconfiguration()	Sends new selected primary subset using a pipe.
		pingreceive_setup()	Sets up the *pingreceive_routine()* sockets and links them to the corresponding network interfaces.
		pingsend_setup()	Sets up the *pingsend_routine()* sockets and links them to the corresponding network interfaces.
	util.c	*delayFile_store()*	Logs the send time, receive time and the RTT of a probe packet to a log file.
		unavFile_store()	Logs unavailability events' intervals to a log file.
		metricsFile_store()	Logs the attributes of the different subsets to a log file.

Daemon association	File name	Function name	Function description
		pathPowerset()	Given a set of paths, the function provides the power set excluding subsets that have a single interface at the source or at the destination or include a number of paths < 2 paths or > 3 paths.
		biCo()	Given a set of n paths, it provides the binomial coefficient representing the number of k subsets that can be obtained out of n.
		subsetselection()	Choses θ_{pr} and θ_{ba} based on (5.13) and (5.14) (the RC4CPS selection criteria). θ_{ba} is selected only if one is available.
Library	**global.c**	*host_store()*	It writes into a file the given host structure which includes the interfaces and their corresponding INDs.
		host_load()	Retrieves the required host structure stored in a file.
		list_init()	Initializes a new list.
		list_append()	Appends an element to an existing list.
		list_delete()	Removes an element from a list.
		list_size()	Returns the number of elements stored in a list.
		list_lock()	Locks a list while being accessed by the calling process to avoid concurrent editing.
		list_unlock()	Unlocks a list for the process calling it.
		iprp_thr_name()	Returns the respective name of a thread for DEBUG message calls.
	receiver.c	*activesenders_store()*	Writes into a file the given active senders.
		activesenders_load()	Loads from a file the active senders list.
	sender.c	*peerbase_store()*	Writes into a file the peer-base structure.
		peerbase_load()	Retrieves from a file the peer-base structure.

List of Tables

List of Figures

Abbreviations

5G	5th Generation
ACK	Acknowledgment
AGC	automatic generation control
AM	Attributes Matrix
AMI	Advanced Metering Infrastructure
AMPTCP	Augmented MPTCP
ANS	Autonomous System Number
API	Application Programming Interface
AS	Autonomous System
ASN	Autonomous System Number
CAN	Controller Area Network
CC	Congestion Control
cc	Correlation coefficient
CCC	Coupled Congestion Control
CDF	Cumulative Distribution Functions
CI	Confidence Interval
CMT/RP-SCTP	Resource Poolingenabled CMT-SCTP
CMT-SCTP	Concurrent Multipath Transfer SCTP
CPS	Cyber-physical System
cTCP	Concurrent TCP
CWAMPTCP	Congestion Window Adaptation MPTCP
DA	Distribution Automation
DARPA	United States Defense Advanced Research Projects Agency
DAR-SCTP	Dynamic Address Reconfiguration SCTP
DBAS	Deployable Bandwidth Aggregation System
DCC	Data and Control Center
DE	Diversity Estimator
DNP3	Distributed Network Protocol 3
DR	Demand Response
DSM	Demand-side management
DSR	Dynamic Source Routing
DTLS	Datagram Transport Layer Security Session
e2e	End-to-end
E2EMPT	End-to-End Multipath Transfer
ECN	Explicit Congestion Notification
EGM	Extended Gilbert model
EtherCAT	Ethernet for Control Automation Technology
FEC	Forward Error Correction
FIP	Factory Instrumentation Protocol
FMTCP	Fountain-Code-Based MPTCP
FPS-SCTP	Forward Prediction Scheduling SCTP
FTP	File Transfer Protocol
G-DBAS	Green DBAS
HMM	Hidden Markov Model
i.i.d.	Independent and identically distributed

IA	Inter-arrival
ICD	iPRP Control Daemon
ICMP	Internet Control Message Protocol
IED	Intelligent Electronic Device
IMD	iPRP Monitoring Daemon
IMMS	Initial Monitoring for Model Selection
IND	iPRP Network subcloud Discriminator
IP	Internet Protocol
IPC	Interprocess Communication
IPD	iPRP Path-selection Daemon
iPRP	The Parallel Redundancy Protocol for IP Networks
iPRP_CAP	iPRP capability message
IRD	iPRP Receiver Daemon
ISD	iPRP Sender Daemon
ISP	Internet Service Provider
LACP	Link Aggregation Control Protocol
LAN	Local Area Network
LG	Looking Glass
LIA	Linked Increases Algorithm
LISP	Locator/Identifier Separation Protocol
LTE	Long-Term Evolution
M&E	Monitoring & Estimation
M/TCP	Multipath Transmission Control Protocol
M2M	Machine-to-Machine
MAP/EPA	Manufacturing Automation Protocol/Enhanced Performance Architecture
MC	Markov Chain
MP	Multipath
MPLS	Multiprotocol Label Switching
MPLOT	Multi-Path LOss-Tolerant
MPRTP	Multipath RTP
MPTCP	Multipath TCP
MPTCP-SPA	MPTCP Slow Path Adaptation
MPTS-AR	Multipath Transport System Based on Application-Level Relay
MTC	Machine-Type Communication
M-TCP	Multipath TCP
MuniSocket	Multiple Network Interface Socket
NAN	Neighborhood-area Network
NASPI	The North American Synchro-Phasor Initiative
NAT	Network Address Translator
NC-MPTCP	Network Coding Based MPTCP
NCC	Network Coordination Centre
NGMN	Next Generation Mobile Networks
NSP	Network service provider
OLIA	Opportunistic Linked Increases Algorithm
OPERETTA	Optimal Energy Efficient Bandwidth Aggregation System
OS	Operating system

OSI	Open System Interconnection
PC	Personal computer
PDU	Protocol data unit
PLC	Programmable logic controller
PMTUD	Path MTU Discovery
PMU	Phasor Measurement Unit
PROFIBUS	PROcessFIeld Bus
PROFINET	Process Field Net
PRP	Parallel Redundancy Protocol
PRR	Ping-Receive Routine
PSN	Probe sequence number
PSockets	Parallel Sockets
PSR	Ping-Send Routines
PSW	Pattern start window
pTCP	Parallel TCP
PUM	Path Usage Method
PW	Pattern window
QoS	Quality of Service
QoS-MPTCP	QoS-oriented MPTCP
RC4CPS	Reliable Multipath Communication for Internet-based CPSs
RI2N/UDP	UDP-based Redundant Interconnection with Inexpensive Network
RIPE	Abbreviated from French for "European IP Networks"
R-M/TCP	Rate-based M/TCP
RMTP	Reliable Multiplexing Transport Protocol
RON	Resilient Overlay Network
RP	Resource Pooling
RT	Real-time
RTP	Real-Time Transport Protocol
RTT	Roundtrip time
RTU	Remote terminal unit
SCADA	Supervisory Control and Data Acquisition
SC-MPTCP	Systematic Coding MPTCP
SCTP	Stream Control Transmission Protocol
SDN	Software-defined Networking
SN	Sequence Number
SNSID	Sequence-Number-Space ID
SR	Selection Routine
SSR	Subflow Sender Reports
TCP	Transmission Control Protocol
TTL	Time-to-Live
UC	Unavailability Calculator
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UP	Unavailability Predictor
VN	Virtual node
WAN	Wide Area Network
WASA	Wide Area Situational Awareness

WAVSM	Wide-area Voltage Stability Monitoring
WRR	Weighted Round-Robin
WSN	Wireless Sensor Network

References

[1] E. A. Lee, "Cyber-physical systems-are computing foundations adequate," in *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, 2006.

[2] J. Gausemeier, R. Dumitrescu, J. Jasperneite, A. Kühn, and H. Trsek, "On the Road to Industry 4.0 - Solutions From the Leading-Edge Cluster It s OWL." it's OWL Clustermanagement GmbH, pp. 1–24, 2014.

[3] IEEE Std 610, "IEEE Std 610 - IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries," *IEEE Std 610*. pp. 1–217, 1991.

[4] C. H. Hauser, D. E. Bakken, and A. Bose, "A failure to communicate: next generation communication requirements, technologies, and architecture for the electric power grid," *IEEE Power Energy Mag.*, vol. 3, no. 2, pp. 47–55, Mar. 2005.

[5] S. Amin, "U.S. grid gets less reliable [The Data]," *IEEE Spectr.*, vol. 48, no. 1, pp. 80–80, Jan. 2011.

[6] D. Babazadeh, M. Chenine, and L. Nordström, "Survey on the Factors Required in Design of Communication Architecture for Future DC grids," *2nd IFAC Workshop on Convergence of Information Technologies and Control Methods with Power Systems, ICPS 2013; Cluj-Napoca; Romania; 22 May 2013 through 24 May 2013*. pp. 58–63, 2013.

[7] A. P. Snow, "Network reliability: the concurrent challenges of innovation, competition, and complexity," *IEEE Trans. Reliab.*, vol. 50, no. 1, pp. 38–40, Mar. 2001.

[8] H. Li, A. Dimitrovski, J. Bin Song, Z. Han, and L. Qian, "Communication Infrastructure Design in Cyber Physical Systems with Applications in Smart Grids: A Hybrid System Framework," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 3, pp. 1689–1708, Jan. 2014.

[9] M. Rausand and A. Høyland, *System reliability theory: models, statistical methods, and applications*, vol. 396. John Wiley & Sons, 2004.

[10] R. Sahner, K. Trivedi, and A. Puliafito, *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. .

[11] K. P. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall, "Improving the reliability of internet paths with one-hop source routing," *Oper. Syst. Des. Implement.*, p. 13, Dec. 2004.

[12] Y. Li, Y. Zhang, L. Qiu, and S. Lam, "SmartTunnel: Achieving Reliability in the Internet," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, 2007, pp. 830–838.

[13] US. DOE, "Communications requirements of Smart Grid technologies," *US Department of Energy, Tech. Rep*. 2010.

[14] B. Carpenter and S. Brim, “Middleboxes: Taxonomy and issues,” 2002.

[15] K. Demir, D. Germanus, and N. Suri, “Robust and real-time communication on heterogeneous networks for smart distribution grid,” in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 386–391.

[16] M. Rentschler and H. Heine, “The Parallel Redundancy Protocol for industrial IP networks,” in *2013 IEEE International Conference on Industrial Technology (ICIT)*, 2013, pp. 1404–1409.

[17] M. Popovic, M. Mohiuddin, D.-C. Tomozei, and J.-Y. Le Boudec, “iPRP: Parallel redundancy protocol for IP networks,” in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, 2015, pp. 1–4.

[18] H. Kirrmann, M. Hansson, and P. Muri, “IEC 62439 PRP: Bumpless recovery for highly available, hard real-time industrial networks,” in *2007 IEEE Conference on Emerging Technologies & Factory Automation (EFTA 2007)*, 2007, pp. 1396–1399.

[19] S. Keshav, *An engineering approach to computer networking: ATM networks, the Internet, and the telephone network*. Addison-Wesley Professional, 1997.

[20] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, “TCP Extensions for Multipath Operation with Multiple Addresses,” *RFC 6824*, 2013.

[21] The MathWorks Inc., “MATLAB.” The MathWorks, Inc., Natick, Massachusetts, United States.

[22] K. Marashi and S. S. Sarvestani, “Towards Comprehensive Modeling of Reliability for Smart Grids: Requirements and Challenges,” *High-Assurance Syst. Eng. (HASE), 2014 IEEE 15th Int. Symp.*, pp. 105–112, 2014.

[23] S. H. Ahmed, G. Kim, and D. Kim, “Cyber Physical System: Architecture, applications and research challenges,” in *2013 IFIP Wireless Days (WD)*, 2013, pp. 1–5.

[24] IEEE, “IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995),” *IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995)*. pp. 1–57, 2006.

[25] IEC, “Communication networks and systems for power utility automation,” p. 10 parts, 2004.

[26] M. Elattar, M. Friesen, D. Henneke, and J. Jasperneite, “Reliability-oriented Multipath Communication for Internet-based Cyber-physical Systems,” in *2018 IEEE World Conference on Factory Communication Systems (WFCS)*, 2018, pp. 1–10.

[27] M. Elattar, T. Cao, V. Wendt, J. Jasperneite, and A. Trächtler, “Reliable Multipath Communication Approach for Internet-based Cyber-physical Systems,” *2017 IEEE 26th Int. Symp. Ind. Electron.*, 2016.

[28] M. Elattar, M. Friesen, and J. Jasperneite, “Evaluation of Multipath Communication Protocols for Reliable Internet-based Cyber-physical Systems,” *2017 IEEE 26th Int. Symp. Ind. Electron.*

[29] M. Elattar, V. Wendt, and J. Jasperneite, “Communications for Cyber-Physical Systems,” in *Industrial Internet of Things*, Springer International Publishing, 2017, pp. 347–372.

[30] D. Henneke, M. Elattar, and J. Jasperneite, “Communication patterns for Cyber-Physical Systems,” in *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, 2015, pp. 1–4.

[31] M. Elattar and J. Jasperneite, “Using LTE as an access network for internet-based cyber-physical systems,” in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, 2015, pp. 1–7.

[32] M. Elattar, L. Dürkop, and J. Jasperneite, “Utilizing LTE QoS features to provide a reliable access network for cyber-physical systems,” in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, pp. 956–961.

[33] M. Elattar, V. Wendt, A. Neumann, and J. Jasperneite, “Potential of multipath communications to improve communications reliability for internet-based cyberphysical systems,” in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–8.

[34] Ericsson Research, “M2M communications in future cellular networks,” 2014. [Online]. Available: https://www.hs-osnabrueck.de/fileadmin/HSOS/Forschung/Recherche/Laboreinrichtungen_und_Versuchsbetriebe/Labor_fuer_Hochfrequenztechnik_und_Mobilkommunikation/Mobilkomtagung/2014/2_Nadia_Brahmi.pdf. [Accessed: 06-Jun-2016].

[35] A. Faza, S. Sedigh, and B. McMillin, “Integrated Cyber-Physical Fault Injection for Reliability Analysis of the Smart Grid,” in *Computer Safety, Reliability, and Security: 29th International Conference, SAFECOMP 2010, Vienna, Austria, September 14-17, 2010. Proceedings*, Springer Berlin Heidelberg, 2010.

[36] C. Singh and A. Sprintson, “Reliability assurance of cyber-physical power systems,” in *IEEE PES General Meeting*, 2010, pp. 1–6.

[37] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, “Communication network requirements for major smart grid applications in HAN, NAN and WAN,” *Comput. Networks*, vol. 67, pp. 74–88, 2014.

[38] N. Dorsch, H. Georg, and C. Wietfeld, “Analysing the real-time-capability of wide area communication in smart grids,” *Proc. - IEEE INFOCOM*, pp. 682–687, 2014.

[39] E. A. Lee, “Cyber Physical Systems: Design Challenges,” in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363–369.

[40] J. Wan, M. Chen, F. Xia, D. Li, and K. Zhou, “From machine-to-machine communications towards cyber-physical systems,” *Comput. Sci. Inf. Syst.*, vol. 10, no. 3, pp. 1105–1128, 2013.

[41] D. Pradhan, “Fault-tolerant computer system design,” 1996.

[42] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr, “Basic concept and

taxonomy of dependable and secure computing," 2004.

[43] N. Edwards, "Building dependable distributed systems," *ANSA, Feb*, 1994.

[44] M. Liotine, *Mission-critical network planning*. Artech House, 2003.

[45] M. Al-Kuwaiti, N. Kyriakopoulos, and S. Hussein, "A comparative analysis of network dependability, fault-tolerance, reliability, security, and survivability," *IEEE Commun. Surv. Tutorials*, vol. 11, no. 2, pp. 106–124, 2009.

[46] J. D. McCabe, *Practical computer network analysis and design*. Morgan Kaufmann Publishers Inc., 1998.

[47] K. Budka, J. Deshpande, and M. Thottan, *Communication Networks for Smart Grids*. Springer, 2014.

[48] M. Dahlin, B. V. Chandra, and A. Nayate, "End-to-end WAN service availability," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 300–313, Apr. 2003.

[49] V. Paxson, "End-to-end routing behavior in the Internet," *Networking, IEEE/ACM Trans.*, 1997.

[50] Y. Zhang, V. Paxson, S. Shenker, and L. Breslau, "The stationarity of internet path properties: Routing, loss, and throughput," 2000.

[51] W. Jiang and H. Schulzrinne, "Assessment of voip service availability in the current internet," *Proc. 4th Int. Work. Passiv. Act. Netw. Meas. (PAM 2003)*, 2003.

[52] A. Shpiner, Y. Revah, and T. Mizrahi, "Multi-path Time Protocols," in *2013 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS) Proceedings*, 2013, pp. 1–6.

[53] F. Jahanian, "Impact of path diversity on multi-homed and overlay networks," in *International Conference on Dependable Systems and Networks, 2004*, 2004, pp. 29–38.

[54] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman, "A measurement-based analysis of multihoming," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*, 2003, p. 353.

[55] I. Stoica and R. H. Katz, "Backup path allocation based on a correlated link failure probability model in overlay networks," in *10th IEEE International Conference on Network Protocols, 2002. Proceedings.*, 2002, pp. 236–245.

[56] J. C. (Jean-C. Laprie, *Dependability : basic concepts and terminology in English, French, German, Italian, and Japanese*. Springer-Verlag, 1992.

[57] S. Savage *et al.*, "The end-to-end effects of Internet path selection," in *Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication - SIGCOMM '99*, 1999, vol. 29, no. 4, pp. 289–299.

[58] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolie, "Design of a low-latency, high-reliability wireless communication system for control applications," in *2014*

IEEE International Conference on Communications, ICC 2014, 2014, pp. 3829–3835.

[59] S.-C. Lin, L. Gu, and K.-C. Chen, "Statistical Dissemination Control in Large Machine-to-Machine Communication Networks," *IEEE Trans. Wirel. Commun.*, vol. 14, no. 4, pp. 1897–1910, Apr. 2015.

[60] D. Niyato, Ping Wang, and E. Hossain, "Reliability analysis and redundancy design of smart grid wireless communications system for demand side management," *IEEE Wirel. Commun.*, vol. 19, no. 3, pp. 38–46, Jun. 2012.

[61] X. Wu, Y. Dong, Y. Ge, and H. Zhao, "A High Reliable Communication Technology in Electric Vehicle Charging Station," in *2013 IEEE Seventh International Conference on Software Security and Reliability Companion*, 2013, pp. 198–203.

[62] H. Ahmadi and T. Abdelzaher, "An Adaptive-Reliability Cyber-Physical Transport Protocol for Spatio-temporal Data," in *2009 30th IEEE Real-Time Systems Symposium*, 2009, pp. 238–247.

[63] I. Lopez, M. Aguado, and E. Jacob, "End-to-End Multipath Technology: Enhancing Availability and Reliability in Next-Generation Packet-Switched Train Signaling Systems," *IEEE Veh. Technol. Mag.*, vol. 9, no. 1, pp. 28–35, Mar. 2014.

[64] I. Lopez, M. Aguado, D. Ugarte, A. Mendiola, and M. Higuero, "Exploiting redundancy and path diversity for railway signalling resiliency," in *2016 IEEE International Conference on Intelligent Rail Transportation (ICIRT)*, 2016, pp. 432–439.

[65] Y. Hu *et al.*, "NASPInet Specification - An Important Step toward Its Implementation," in *2010 43rd Hawaii International Conference on System Sciences*, 2010, pp. 1–9.

[66] L. Wisniewski, M. Schumacher, J. Jasperneite, and C. Diedrich, "Linear time, possibly disjoint path search approach for ethernet based industrial automation networks," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, pp. 1–9.

[67] K. J. Park, J. Kim, H. Lim, and Y. Eun, "Robust path diversity for network quality of service in cyber-physical systems," *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2204–2215, Nov. 2014.

[68] M. E. Tozal, E. Al-Shaer, K. Sarac, and B. Thuraisingham, "On secure and resilient telesurgery communications over unreliable networks," in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 714–719.

[69] A. Lukyanenko, "Network coding based multipath TCP," in *2012 Proceedings IEEE INFOCOM Workshops*, 2012, pp. 25–30.

[70] J. Liao, J. Wang, T. Li, and X. Zhu, "Introducing multipath selection for concurrent multipath transfer in the future internet," *Comput. Networks*, vol. 55, no. 4, pp. 1024–1035, 2011.

[71] H. Haddadi and O. Bonaventure, *Recent advances in networking*. ACM SIGCOMM, 2013.

[72] ITU, "The Tactile Internet," 2014. [Online]. Available: http://www.itu.int/en/ITU-T/techwatch/Pages/tactile-internet.aspx. [Accessed: 12-May-2016].

[73] S. Habib, J. Qadir, A. Ali, D. Habib, M. Li, and A. Sathiaseelan, "The past, present, and future of transport-layer multipath," *J. Netw. Comput. Appl.*, vol. 75, pp. 236–258, 2016.

[74] M. Li, A. Lukyanenko, Z. Ou, and A. Yla-Jaaski, "Multipath transmission for the internet: A survey," *Tutorials, vol. PP*, 2016.

[75] M. Li, "Improving the Efficiency of Multipath Transport Protocols," Aalto University, 2014.

[76] K. Habak, K. A. Harras, and M. Youssef, "Bandwidth aggregation techniques in heterogeneous multi-homed devices: A survey," *Comput. Networks*, vol. 92, pp. 168–188, 2015.

[77] V. Jacobson, R. Frederick, S. Casner, and H. Schulzrinne, "RTP: A Transport Protocol for Real-Time Applications."

[78] K. Demir, D. Germanus, and N. Suri, "Robust and real-time communication on heterogeneous networks for smart distribution grid," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 386–391.

[79] Kurose and Ross, *Computer Networking - A Top-Down Approach*. Pearson Education, 2013.

[80] M. Dye, R. McDonald, and A. Rufi, *Network Fundamentals, CCNA Exploration Companion Guide*. Cisco Press, 2007.

[81] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach: International Edition*. Pearson Higher Ed, 2013.

[82] ETR003 ETSI, "Network Aspects (NA); General Aspects of Quality of Service (QoS) and Network Performance (NP)," *Technical Report*. 1994.

[83] ITU-T, "Terms and definitions related to quality of service and network performance including dependability," *Recommendation E 800*. 1994.

[84] E. Kamrani, H. Momeni, and A. Sharafat, "Modeling internet delay dynamics for teleoperation," in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, 2005, pp. 1528–1533.

[85] B. Wilamowski and J. Irwin, *Industrial communication systems*. CRC Press, 2011.

[86] Y. Tipsuwan and M. Chow, "Control methodologies in networked control systems," *Control Eng. Pract.*, vol. 11, pp. 1099--1111, 2003.

[87] R. Khan and J. Khan, "A comprehensive review of the application characteristics and traffic requirements of a smart grid communications network," *Comput.*

Networks, 2013, vol. 57, pp. 825--845, 2013.

[88] D. Boswarthick, O. Elloumi, and O. Hersent, *M2M Communications: A Systems Approach*. John Wiley & Sons, 2012.

[89] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *PROCEEDINGS-IEEE*, vol. 95. p. 138, 2007.

[90] J. Aho *et al.*, "A tutorial of wind turbine control for supporting grid frequency through active power control," in *2012 American Control Conference (ACC)*, 2012, pp. 3120–3131.

[91] The 3rd Generation Partnership Project (3GPP), "3GPP Release 99," 1999. [Online]. Available: http://www.3gpp.org/. [Accessed: 10-May-2016].

[92] Open Networking Foundation, "Software-defined networking: The new norm for networks," *ONF White Paper*. 2012.

[93] J. Abbate, "Building the Arpanet: Challenges and Strategies," in *Inventing the Internet*, 1999.

[94] K. Fall and W. Stevens, *TCP/IP illustrated, volume 1: The protocols*. Addison-Wesley Professional, 2011.

[95] A. Medina, M. Allman, and S. Floyd, "Measuring interactions between transport protocols and middleboxes," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement - IMC '04*, 2004, p. 336.

[96] D. Borman, R. Scheffenegger, and V. Jacobson, "TCP extensions for high performance," 2014.

[97] IANA (Internet Assigned Numbers Authority), "Internet Protocol Version 4 (IPv4) Parameters." [Online]. Available: http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml. [Accessed: 21-Feb-2017].

[98] W. Noonan and I. Dubrawsky, "How Broadband Routers and FirewallsWork," in *Firewall Fundamentals*, Cisco Press, 2006.

[99] T. W. Shinder, *Best damn firewall book period*. Syngress, 2007.

[100] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," *Internet Engineering Task Force, RFC3168*. 2001.

[101] J. Mogul and S. Deering, "Path MTU discovery," *Internet Engineering Task Force, RFC1063*. 1990.

[102] P. Srisuresh and K. Egevang, "Traditional IP network address translator (Traditional NAT)," *Internet Engineering Task Force, RFC1631*. 2000.

[103] L. D'Acunto, J. Pouwelse, and H. Sips, "A measurement of NAT and firewall characteristics in peer-to-peer systems," *Proc. 15-th ASCI Conf.*, 2009.

[104] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 2, pp. 860–880, Jan. 2013.

[105] IEC, "Digital data communications for measurement and control – Fieldbus for use in industrial control systems." IEC Std. 61158, 1999.

[106] K. Curtis, “A DNP3 Protocol Primer,” *DNP User Gr.*, pp. 1--8, 2005.

[107] IEEE, “IEEE Standard for Electric Power Systems Communications – Distributed Network Protocol (DNP3),” *IEEE Std 1815-2010*. IEEE Std 1815-2010, 2012.

[108] NGMN Alliance, “NGMN 5G White Paper,” 2015. [Online]. Available: http://ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf. [Accessed: 12-May-2016].

[109] J. Tsai and T. Moors, “A review of multipath routing protocols: From wireless ad hoc to mesh networks,” *Res. Work. Wirel. multihop Netw.*, 2006.

[110] S. Habib, J. Qadir, A. Ali, D. Habib, M. Li, and A. Sathiaseelan, “The past, present, and future of transport-layer multipath,” *J. Netw. Comput. Appl.*, vol. 75, pp. 236–258, 2016.

[111] M. Li, A. Lukyanenko, Z. Ou, and A. Yla-Jaaski, “Multipath transmission for the internet: A survey,” *Tutorials, vol. PP*, 2016.

[112] J. Qadir, A. Ali, K.-L. A. Yau, A. Sathiaseelan, and J. Crowcroft, “Exploiting the Power of Multiplicity: A Holistic Survey of Network-Layer Multipath,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2176–2213, 2015.

[113] D. Wischik, M. Handley, and M. B. Braun, “The resource pooling principle,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 5, p. 47, Sep. 2008.

[114] U.S. Department of Energy (DOE), “The Smart Grid: An Introduction.” 2008.

[115] US. DOE, “Communications Requirements of Smart Grid Technologies,” *US Department of Energy, Tech. Rep.* 2010.

[116] P. Sahu and M. K. Verma, “On-line voltage stability monitoring and control in smart grid — A survey,” *2015 IEEE UP Sect. Conf. Electr. Comput. Electron.*, no. December 2015, pp. 1–6, 2015.

[117] M. Glavic and D. Novosel, “Voltage Stability Monitoring , Instability Detection and Control,” *Quanta Technology, Tech. Rep.*, 2011. .

[118] ABB Switzerland Ltd, “A PSGuard Wide Area Monitoring System application.”

[119] M. L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*. John Wiley & Sons, 2003.

[120] G. Sandler, *System reliability engineering*. Englewood Cliffs, N.J.: Prentice-Hall, 1963.

[121] C. H. Lie, C. L. Hwang, and F. A. Tillman, “Availability of Maintained Systems: A State-of-the-Art Survey,” *A I I E Trans.*, vol. 9, no. 3, pp. 247–259, Sep. 1977.

[122] D. Rubenstein, J. Kurose, and D. Towsley, “Detecting shared congestion of flows via end-to-end measurement,” *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 381–395, Jun. 2002.

[123] A. Konrad, B. Y. Zhao, and A. D. Joseph, “Determining model accuracy of network traces,” *J. Comput. Syst. Sci.*, vol. 72, no. 7, pp. 1156–1171, Nov. 2006.

[124] H. S. Wenyu Jiang, “Modeling of Packet Loss and Delay and their Effect on Real-Time Multimedia Service Quality,” *PROCEEDINGS OF NOSSDAV ’2000*. .

[125] M. Yajnik, J. Kurose, and D. Towsley, “Packet loss correlation in the MBone multicast network,” *Technical report*. pp. 94–99, 1996.

[126] “PlanetLab.” [Online]. Available: www.planet-lab.org.

[127] “NorNet.” [Online]. Available: www.nntb.no.

[128] S. Sanfilippo, “HPing home page.” .

[129] “RIPE Network Coordination Centre.” [Online]. Available: www.ripe.net.

[130] R. Steenbergen, “A practical guide to (correctly) troubleshooting with traceroute,” 2009. [Online]. Available: http://newnog.biz/sites/default/files/tuesday_steenbergen_troublshootingtraceroute_62.49.pdf. [Accessed: 06-Jun-2016].

[131] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, “Measuring ISP Topologies With Rocketfuel,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 2–16, Feb. 2004.

[132] B. Augustin, T. Friedman, and R. Teixeira, “Measuring load-balanced paths in the internet,” in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement - IMC ’07*, 2007, p. 149.

[133] M. Ellis, D. P. Pezaros, T. Kypraios, and C. Perkins, “A two-level Markov model for packet loss in UDP/IP-based real-time video applications targeting residential users,” *Comput. Networks*, vol. 70, pp. 384–399, 2014.

[134] A. Konrad and A. D. Joseph, “Choosing an accurate network path model,” *Rep. No. UCB/CSD-03-1236*. 2003.

[135] H. A. Sanneck and G. Carle, “A Framework Model for Packet Loss Metrics Based on Loss Runlengths,” *Proc. SPIE 3969, Multimed. Comput. Netw. 2000, 177 (December 27, 1999)*, vol. 3969, no. January 2000, pp. 177–187, 2000.

[136] M. Yajnik, J. Kurose, and D. Towsley, “Measurement and modelling of the temporal dependence in packet loss,” in *IEEE INFOCOM ’99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320)*, 1999, vol. 1, pp. 345–352 vol.1.

[137] E. N. Gilbert, “Capacity of a Burst-Noise Channel,” *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, Sep. 1960.

[138] L. Finesso, Chuang-Chun Liu, and P. Narayan, “The optimal error exponent for Markov order estimation,” *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1488–1497, 1996.

[139] N. Merhav, M. Gutman, and J. Ziv, “On the estimation of the order of a Markov chain and universal data compression,” *IEEE Trans. Inf. Theory*, vol. 35, no. 5, pp. 1014–1019, 1989.

[140] Shu Tao and R. Guerio, “On-line estimation of internet path performance: an application perspective,” in *IEEE INFOCOM 2004*, vol. 3, pp. 1774–1785.

[141] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, “A Markov-based channel model algorithm for wireless networks,” in *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems - MSWIM ’01*, 2001, pp. 28–36.

[142] B. Hesmans and O. Bonaventure, “An enhanced socket API for Multipath TCP,” in *Proceedings of the 2016 workshop on Applied Networking Research Workshop - ANRW 16*, 2016, pp. 1–6.

[143] R. Stewart, M. Tuexen, K. Poon, P. Lei, and V. Yasevich, “Sockets API extensions for the stream control transmission protocol (SCTP),” 2011.

[144] C. Raiciu, M. Handley, and D. Wischik, “Coupled Congestion Control for Multipath Transport Protocols,” *RFC 6356*, 2011.

[145] R. Khalili, N. Gast, and M. Popovic, “MPTCP is not pareto-optimal: performance issues and a possible solution,” *IEEE/ACM Trans.*, 2013.

[146] Information Sciences Institute ISI, “The Network Simulator - ns-2.” 2016.

[147] L. Ottet and M. Mohiuddin, “iPRP Repository.” [Online]. Available: https://github.com/Brebiche38/IPRP.

[148] Globus Alliance, “GT 6.0 GridFTP.” 2016.

[149] B. Kreith, V. Singh, and J. Ott, “Multipath-RTP Repository.” 2016.

[150] NIST/SEMATECH, *e-Handbook of Statistical Methods*. .

[151] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. Wiley, 2011.

[152] C. Geyer, “Stat 5102 Notes: Nonparametric Tests and Confidence Intervals.” 2003.

[153] PacketStorm Communications Inc., “PacketStorm1800E IP network emulator.” .

[154] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, “Measurement and analysis of single-hop delay on an IP backbone network,” *IEEE J. Sel. Areas Commun.*, vol. 21, no. 6, pp. 908–921, Aug. 2003.

[155] J. Postel, “User Datagram Protocol,” *RFC 768 (INTERNET Stand.*, 1980.

[156] Miroslav Popovic, “Redundancy in Communication Networks for Smart Grids,” Swiss Federal Institute of Technology in Lausanne, 2016.

[157] The Netfilter Core Team, “The netfilter.org ‘libnetfilter_queue’ project.” 2017.

[158] S. Hemminger, “Network Emulation with NetEm,” *Proc. 6th Aust. Natl. Linux Conf. (LCA 2005)*, no. April, pp. 1–9, 2005.

[159] ITU-T, “Network model for evaluating multimedia transmission performance over Internet Protocol,” 2007.

[160] M. Popovic, M. Mohiuddin, D. Tomozei, and J. Le Boudec, “iPRP : Parallel

Redundancy Protocol for IP Networks,” 2015.